AIAA FLIGHT SIMULATION TECHNOLOGIES CONFERENCE AND EXHIBIT

A COLLECTION OF TECHNICAL PAPERS
BOSTON, MASSACHUSETTS
AUGUST 14–16, 1989

AIAA

For permission to copy or republish, contact:
The American Institute of Aeronautics and Astronautics
370 L'Enfant Promenade, SW
Washington, DC 20024-2518

A COLLECTION OF TECHNICAL PAPERS

AIAA FLIGHT SIMULATION TECHNOLOGIES CONFERENCE AND EXHIBIT

AUGUST 14–16, 1989 / BOSTON, MASSACHUSETTS

Printed in the U.S.A.

AIAA Flight Simulation Technologies Conference and Exhibit
August 14-16, 1989 / Boston, MA

Table of Contents

Paper No. **Title and Author** **Page No.**

SESSION 1 - Verification and Validation

SESSION 2 - Space Simulation

SESSION 3 - Simulator Sickness and Cue Syncronization

SESSION 4 - Motion Cueing

W-Withdrawn
NA-Not Available

SESSION 5 - Simulation Languages

SESSION 6 - Modeling

SESSION 7 - Visual System I

SESSION 8 - Crew Station Design

W-Withdrawn
NA-Not Available

SESSION 9 - Computer Systems

SESSION 10 - Training Systems

SESSION 11 - Computational Methods

SESSION 12 - Air Combat Simulation

W-Withdrawn
NA-Not Available

W- Withdrawn
NA-Not Available

Paper No. 89-3260, AIAA Flight Simulation Technologies Conference
14-16 August 1989, Boston, MA

89-3260-CP

A COMPREHENSIVE COLLECTION OF PROCEDURES FOR SIMULATION VERIFICATION

Matthew S. Middendorf
William V. Johnson
Michael J. Gilkey
Terrence D. McClurg

Systems Research Laboratories, Inc.
A Division of Arvin Calspan
Dayton , OH 45440

Abstract

Because simulators are becoming more widely used for training and research, it is becoming increasingly important that the simulators be verified to ensure that they behave as designed and intended. The incorporation of new technologies has made simulators more complex which has forced the verification process to become more diverse and thorough.

The dynamic models used in the simulators have typically been the focus of the verification process. Discussing the verification of highly sophisticated models can be a whole paper in itself. This paper will touch upon the verification of simple models, but will concentrate on a wide range of other simulator components that should be verified. This list of components and procedures results from nearly a decade of experience in building research simulators.

Although this list of components was compiled while developing flight simulators used in research laboratories, its applicability is not limited to those simulators. Components such as timing integrity, communication validity, and input control scaling are important to all real-time simulators. Procedures for verifying these components, and several others, are covered in this paper.

Additional components include motion loop delay measurement, visual loop delay measurement, graphics equations, forcing function spectra and filtering, motion cuing device scaling, time tags on collected data, ramp functions, numerical integration, the Euler transformation matrix, and the Euler equations of motion. Each one of these components must be verified through some quantitative means.

Introduction

It is paramount that research simulators be verified to ensure the validity of the collected data. If a long term research effort is conducted in a simulator that contains a mistake, then two serious consequences may result. First, if the experimental data are used to guide simulator design and development, then the simulator industry may be lead astray. Secondly, if the mistake is uncovered at or near the end of a study, then considerable time and money is wasted. To reduce the likelihood of mistakes, a comprehensive set of rigorous procedures needs to be meticulously followed. The procedures described in this paper are mainly quantitative in nature, however, subjective evaluation by an experienced individual also proves to be quite valuable.

Recently a test pilot flew one of the laboratory's fighter-type aircraft simulators and commented that the dynamics seemed to be biased to the left. His specific comment about the dynamics was proven to be incorrect, but in flying the simulator the bias was apparent. This led to investigation of other simulator components, specifically the input controller (in this case a side-mounted force stick). It was discovered that the input controller was rotated seven degrees off-axis. This finding led to an improvement to the simulator and an addition to the simulator verification procedures. Although subjective comments may also be solicited, they are not a substitute for the objective procedures discussed in this paper.

Components

Timing Integrity

Several simulators used in research incorporate wind gust models and/or sum-of-sines forcing functions to gain spectral information about the human operator and the man-machine system.[1] In such simulators it is very important to have stable time frames to prevent smearing in the frequency domain. Multi-tasking operating systems can cause frame time instability. Several of these systems have a job scheduler that runs at 60 Hertz. The interrupt for the job scheduler could coincide with an interrupt from a programmable clock that is used to drive the real-time simulation software. If this occurs then there will be some jitter in the initiation of the simulation time frames (see figure 1). This instability can reduce the signal-to-noise ratio of data that is already troubled with noise due to the nonlinear human in the loop. Some of the problems associated with multi-tasking operating systems can be avoided by turning off some system services, and in some cases, disabling the system clock.

Using the exact duration of the time frame in the calculation of coefficients and integration constants can be as important as having stable timing . If the time frame is off by just a little (59.8 Hertz compared to 60.0), then the characteristics of the dynamics, the integrated values, and the overall duration of the trial (for experiments that use fixed length trials) will be slightly in error. This problem is most prevalent in multi-computer simulations where the computers need to be synchronized, particularly when there is a separate graphics computer. One way to synchronize these computers is to detect the vertical retrace of the graphics monitor and use this signal to drive the other computers. If this method is used then the exact duration of the time frame can be measured using a programmable clock. This measurement should be taken every time the simulator is powered up. Two graphics computers made by the same vendor can have slightly different refresh rates. Also, switching between the two major image presentation formats (60 Hertz non-interlaced and 30 Hertz interlaced) can have a significant impact on the duration of the time frame. Specifically, some 30 Hertz interlaced systems actually have vertical retraces occurring at 66 Hertz. A digital output bit and an oscilloscope can be used to verify both the simulation time frame stability and duration (see figure 1).

Each time frame can be initiated by an interrupt from a programmable clock or by an interrupt from an external input, preferably from the vertical retrace of the graphics monitor. In both cases the digital output bit can be set when the interrupt first occurs and cleared after all software executes. Displaying the value of the digital output on an oscilloscope will yield both the exact duration of the time frame and the execution time of the software. To observe frame time stability it may be necessary to put two full frames up on the scope because the scope may automatically trigger off the setting of the bit thus eliminating the jitter on the first frame. However, if there is instability then it can be seen in the start of the second frame (see figure 1).

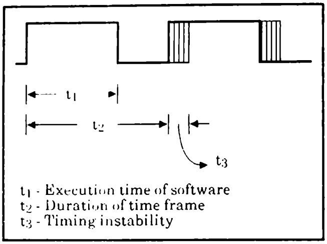

t_1
t_2
t_3
t_1 - Execution time of software
t_2 - Duration of time frame
t_3 - Timing instability

Figure 1. Frame timing and instability

The execution time of individual routines is also easily measured using this setup. This can be achieved by clearing the bit before the particular routine and setting the bit after the routine (see figure 2).

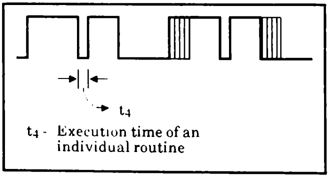


Figure 2. Execution time measurement

Another aspect of timing integrity involves the verification of ramp times. Very often ramps are used to ensure smooth initiation of motion cuing, injection of independent variables, and introduction of gusts. If the subject/pilot is being scored based on his or her performance in the simulator, it is important that the ramps be completed before scoring begins (see figure 3). This can be verified by hand-checking the ramp time constants and comparing them to the time when scoring begins. Another way to verify the ramps is to review time history plots of the pertinent variables from a collected data file. If the ramps are not properly synchronized it could effect the scoring algorithm, bias the experimental variables, and decrease signal-to-noise ratios if frequency domain analysis is being used.

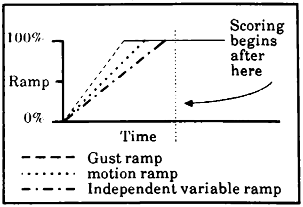


Figure 3. Simulation ramps

When time history files are collected it is important to verify the time tags (i.e. the effective sampling instance) for each channel of data. In most cases the time tags should match each other, however, in some experiments the investigator may request that the tags mismatch for analytical purposes. Improperly matched channels are particularly problematic when performing frequency domain analysis. An incorrect time tag will manifest itself as a phase shift when that particular channel is used to form a transfer function. A simple example using a difference equation of the form $Y = (X_n - X_{n-1})/T$ for numerical differentiation (see figure 4) will illustrate how two collected channels of data can have mismatched time tags. This equation fits the form of a finite impulse response filter (FIR) and has a signal time delay associated with it of T/2. [2] Another example where a mismatch is an artifact of the specific numerical technique is the use of an Adams-Bashforth integration algorithm. The use of this algorithm can introduce a time advance relative to the data collection time reference. [3] Additionally, feedback signals implemented in a sampled data system introduce delays due to the order of operations that must be accounted for. Data channel time tags can be checked by time history examination of channels after driving the system with a known forcing function. Frequency analysis of the same type of data will reveal any mismatches by phase deviations from the expected values. These same considerations affect the validation of transport delay to be covered later.

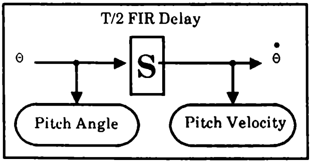


Figure 4. Numerical delay

Another timing integrity issue is the stationary initiation of data collection. This consideration applies specifically to research simulators where the trial is initiated by an experimenter. If the trial beginning and subsequent data collection are dependent on a button being pushed or a switch being thrown then all resultant actions by the software must be keyed off the same moment. Specifically, when the button is pressed or released, but not both. This is particularly important when time history data is being averaged. Verification of stationary data collection is simple when there is a deterministic data channel (i.e. a gust channel computed using a sum-of-sines technique). Examination of the first data point in the gust channel across several files will accomplish this task.

The last timing integrity issue is the verification of delay queues. In some cases vehicle states (attitude and position) are delayed before they are sent to the motion cuing device so that the motion information presented to the pilot can be synchronized with the visual information presented by the graphics system. In other cases, the opposite is true. In some research simulators the delay in the visual loop is manipulated as the independent variable therefore requiring a delay queue. If delay queues are implemented it is important to verify that they are producing the correct amount of delay. Care must be taken to see if the queue is updated before or after the delayed value is pulled out to prevent a one frame delay error. This can be accomplished by inspecting the delay queue logic and code. The motion loop and visual loop delay measurement sections of this paper will present a quantitative verification procedure.

Vehicle Dynamics Verification

As stated earlier, it is not the purpose of this paper to deal with the verification of sophisticated full nonlinear aerodynamic models. The models used in the research laboratory where these procedures have evolved have been, for the most part, transfer function models. Verification of these models is a straight forward process utilizing a linear autopilot and a sum-of-sines approximation to a Dryden [4] wind gust model (see figure 5). The simulation is run, with the autopilot engaged, and the collected data is analyzed using frequency domain techniques. The resultant frequency data for the inputs and outputs of each element in the aerodynamic model are used to form transfer functions. For each transfer function the experimental gain and phase information is computed. This experimental data is then compared to analytical data for verification of the aerodynamic model.

Visual Loop Delay Measurement

Research has shown that excessive transport delay in aircraft simulators can have negative effects on pilot performance and transfer of training.[5] Therefore it is very important to be able to accurately quantify the transport delay in simulators. Moreover, the verification of the actual delay against the expected value is an excellent check of other simulator components. The transport delay is defined as the delay due to the computer implementation of the simulation and does not include delay due to the dynamics of the simulated vehicle.[6] This delay is measured between pilot control input and pilot cuing. The measurement is accomplished using a frequency domain steady state technique. A sine wave is injected into the system in place of the input controller and a photo-sensitive device is used to measure resultant changes on the graphics monitor. Using a frequency response analyzer, the total phase shift is measured from the sine wave input to the photo-sensitive device output (see figure 6). Then the phase shift due to the dynamics is subtracted from the total phase shift and the remaining phase shift is due to the transport delay. Because the input sine wave is at a known frequency, the phase shift can be converted to time, in seconds. Consider the following equation:

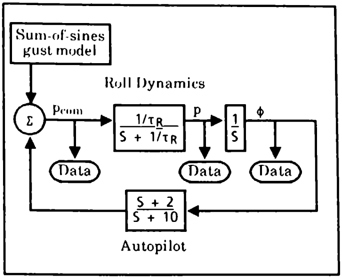


Figure 5. Simplified Roll Dynamics

Transport delay(seconds) =
(total measured phase(Deg) -
Vehicle dynamics phase(Deg)) *
(Period of input frequency/360) (1)

The empirically derived transport delay measurement is then compared to the analytically derived value. If they do not match then the simulation may contain an error, or, a component may not be accounted for in the analytical computation. Some sources of transport delay are execution time, delay queues, numerical techniques, sample and hold (zero order hold), graphics processing, and delay due to order of operations.

Motion Loop Delay Measurement

For motion-based simulators, the motion loop is the path shown in the block diagram , Figure 7, from the control input to the motion display. The transport delay around the motion loop is a simulation parameter that must be characterized and controlled. This delay must be theoretically determined and experimentally verified.

The sources of transport delay around the motion loop are three: 1) delay due to the order of I/O operations; 2) zero order hold delay of the digital-to-analog converters (D/A's); and 3) motion display device servocontrollers. The delay due to order of operations occurs when the A/D's are sampled before the D/A's are output. This is commonly done to give the D/A's maximum settling time before sampling the A/D's and to preserve the periodicity of the I/O operations (e.g. due to variable execution time). Thus an artificial delay equal to one simulation frame time is incurred. A zero order hold delay is an artifact of the reconstruction of an analog signal from a digital signal resulting in an effective delay of $\frac{1}{2}$ of

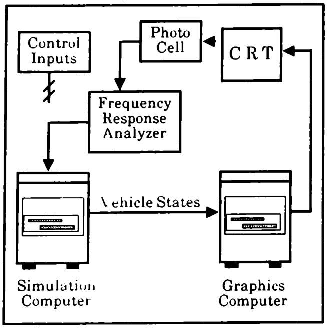


Figure 6. Visual Loop Delay Measurement Setup

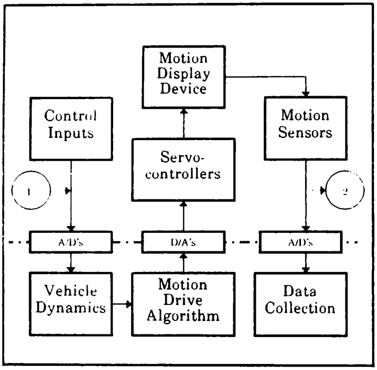


Figure 7. Motion Loop Delay Measurement Setup

the frame time. Last of all, the servocontrollers can be expected to include some signal conditioning electronics, such as low-pass filters, which affect the phase response of the motion device. The frequency characteristics of the servos are usually specified by the manufacturer or can be analytically determined and should be measured empirically as well.

With the above information, the theoretical transport delay around the motion loop can be determined. Once this has been done, the actual system must be verified by comparison with experimentally collected data. Using a frequency response analyzer, a sinusoid at a known frequency can be injected into the loop at point 1 in Figure 7 above. The frequency must be within the bandwidth of the system in order to achieve meaningful results. By feeding back the system response at point 2 to the analyzer, the gain and phase of the system between the two points is computed. The total delay around the loop at that frequency is given by the following equation:

$$\Delta t\ (sec) = \phi\ (rad) \div f\ (rad/sec) \qquad (2)$$

Time delays due to the vehicle dynamics and drive algorithms are not transport delays and must be subtracted from these experimental measurements. Care must be taken to account for 180° phase shifts due to sign changes in the dynamics or drive algorithm.

Motion Device Scaling

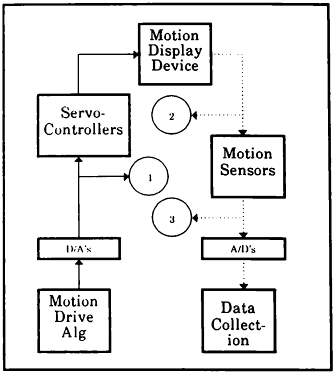


Figure 8. Motion Device Scaling Setup

Figure 8 shows a simplified block diagram encompassing the motion display device and its associated control and recording functions. The motion drive algorithm computes the type and amount of desired platform movement. This quantity is scaled appropriately and output to a digital to analog converter (D/A). The D/A outputs an analog voltage proportional to the digital input. This signal serves as the input to the servocontroller. The servo drives the motion device according to its design.

Motion device scaling is simply the process of verifying the calibration of the blocks between the drive algorithm and the display device. This ensures that the subject is experiencing the desired motion. Another related step which is often overlooked is to verify the calibration of the blocks between the motion device and data collection point. This insures that the movement of the platform is being accurately recorded.

There are two scaling steps required to achieve the desired platform motion from the drive algorithm block. First, the proper servo input voltage must be determined that will result in the desired movement. Next, the digital value for the D/A must be computed that will generate this voltage. The scaling of each motion actuator must be verified individually.

The following equations illustrate this scaling using an example of linear displacement as the measure of motion.

$$D/A\ Input\ (volts) - F_1(Desired\ Displacement) \quad (3)$$

$$Actual\ Displacement(mm) = F_2(Output\ from\ D/A) \quad (4)$$

The only assumption that has been made is that the system is stationary. This simply means that the scale factors do not change with time.

The overall motion scaling is driven by the calibration of the servocontroller. This is an electronic control system with feedback. The scale factor that converts input voltage to platform movement must be verified empirically using a precision voltage source and the appropriate measuring tool. The voltage source is used to inject an arbitrary voltage into a servo as shown at point 1 in Figure 8. The resulting travel of the individual actuator is then measured using a ruler, protractor, or similar device at point 2. Data is collected at regular intervals within the active range of the servo. A least squares fit is then performed on the data to determine the equation of the curve that yields the closest match. This often proves to be linear or nearly so, but even nonlinear relationships can be easily accounted for with software. A highly nonlinear fit may indicate a hardware problem with the servo or motion device and should be investigated further. The best fit relationship that results is the function F_2 of Equation (4).

The next scale factor to be verified is the one that converts the desired platform movement into the proper digital value to send to the D/A. The D/A's are characterized by the number of bits of resolution and analog voltage range. The desired coefficient is simply the number of volts per bit. For example, a twelve bit D/A can represent 4096 distinct levels. If the output range is ± 10.0 volts, there are (20 volts $\div$ 4096=) 4.88 millivolts per level. Therefore, function F_1 of Equation (3) is simply a multiplicative constant possibly with an offset as shown in the example equation (5). This scale factor should be checked regularly to detect hardware problems with the digital-to-analog converter.

$$\text{D/A Input} = 0.00488(\text{volts}^{-1})\ \text{x}$$
$$F_2(\text{desired displ (mm)})(\text{volts}) + \text{Dig Offset} \quad (5)$$

The second part of motion device scaling is to verify the accurate scaling of motion sensor data to real world quantities for data collection purposes. This is simply a matter of calibrating the transducers used in the device as well as the A/D's. The following equations illustrate this scaling.

$$\text{A/D Input (volts)} = \text{F3 (Actual Displacement)} \quad (6)$$

$$\text{Recorded Displacement (mm)} =$$
$$F_4\ (\text{Output from A/D}) \quad (7)$$

For every type of motion there is an appropriate transducer that allows the platform movement to be recorded. In the example of linear displacement used above, a linear variable differential transformer is used (LVDT). This device produces an analog voltage directly proportional to the LVDT shaft displacement and its characteristics are specified by the manufacturer. The calibration of these devices must be verified regularly. This is done by collecting data on the transducer output voltage as the platform moves. This can be done simultaneously with data collection from the first part of motion scaling. While a voltage is being injected into the system at point 1 of Figure 8 and platform motion data is being collected at point 2, the transducer voltage should also be recorded at point 3. Data analysis similar to that used in part 1 will yield the relationship F_3 in equation (6) above.

A/D scaling is the inverse of the D/A scaling described above. Again, this scale factor should be checked regularly for hardware problems.

Communication Validation

The communication link between the simulation computer and the graphics computer must be verified to ensure the integrity of all transmitted data. This is especially important if position deltas are being sent to and summed in the graphics computer. The use of position deltas may be necessary due to hardware limitations (i.e. RS232 updated at 60Hz). The communication link can be easily verified by sending the final position values back to the simulation computer to be compared with its final values. Another aspect of communication that should be verified involves high speed parallel transmission of data. When floating point values are passed it is important to verify that the data representation formats are the same. If not, then the conversion process needs to be verified using a wide range of values including boundary conditions. All data packets should include a checksum.

Verification of Graphics Equations

The order of translations and rotations in the graphics computer needs to be verified. When the vehicle is maneuvered through large angles, errors in the graphics equations are obvious. However, for tasks that require small perturbations about zero there could be an error in the equations that may go unnoticed. Another important graphics parameter that must be verified is the field-of-view (FOV). Conventionally the value used for the field-of-view closely matches the actual hardware setup. Specifically this value equals twice the inverse tangent of the distance from the eye to the screen/crt divided by the half-width of the screen/crt. Care should also be taken to see which axis the field-of-view is with respect to. Some modern graphics computers specify the field-of-view in the y-axis (vertical) while the measurement is taken with respect to the x-axis. For graphics systems that have a four to three (X to Y) aspect ratio, the field-of-view in the x-axis can be transformed to the y-axis using the following equation.

$$FOV_y = TAN^{-1}(TAN(FOV_x)*{}^{3}/_{4}) \quad (8)$$

Verification of Gust Spectra and Gust Filters

This verification procedure is specific to simulations that use gust, or turbulence, that is approximated by a sum-of-sine waves. The use of a sum-of-sines model to approximate a wind gust is well suited for research simulators because the gust inputs appear to be stochastic to the subject and it is useful for frequency domain analysis of the human operator and the man-machine system.[7] If each component in the sum-of-sines profile is an integer harmonic of the fundamental frequency (e.g. the reciprocal of the run length) and is generated without distortion, then the collected data can be analyzed using a fast-Fourier transform (FFT). The results of the FFT can be used to verify the gust spectra. The FFT of the gust spectra should contain power only at the nominal input frequencies.[7] However, all of the FFT data should be examined to make sure there is no significant power at non-input frequencies due to harmonic distortion. Significant power is power above the nominal remnant level due to sampling. Normally when a sum-of-sines profile is selected to model gust there is a desired RMS level for the signal. For example, if a moderate vertical gust is needed to perturb the pitch states of a simulated aircraft then a RMS level of 7.7 feet/second might be selected. The desired RMS level of the input signal can be verified using the FFT data. The square root of the total power computed from the FFT data in the input signal should equal the selected RMS value.

In some cases the gust input is filtered to reflect the effect of the vehicle's immersion in the gust medium. Secondly, the gust may be filtered to determine its effect on the vehicle. The verification of these filters can be easily accomplished using the FFT data. The FFT data can be used to compute a transfer function for each of the filters. The characteristics of the transfer function can be compared against the design specifications of the filters for verification.

Controls Scaling and Deadband Verification

All flight simulations require user inputs of some type to allow control of the simulated aircraft. These can be control sticks, throttles, or instrumentation switches. This paper will concentrate on the control stick and throttle, since these are the primary devices that close the man-machine loop.

Control sticks can be either of a displacement or force type. Both are transducers which generate an output voltage proportional to stick movement or applied stick force. Since only side-mounted force sticks are used here in the laboratory where these procedures evolved, the following discussion will concentrate on these. The comments on force sticks can be directly extended to displacement sticks as well. The hardware and software issues of control scaling will be discussed here.

The stick has both roll and pitch rate control commanded with forces along the lateral and longitudinal axes, respectively. The device must be checked to insure that these axes are orthogonal. This is done by applying a force along one stick axis and reading the voltage of the off-axis output with a voltmeter. The force should be rotated about the stick's vertical axis until the output voltage is nulled. The direction of the force vector should be marked and the step repeated for the opposite direction. The positive and negative directions of each axis should be co-linear. This can then be done for the other axis. The two axes should be orthogonal.

Now that the control device axes have been characterized, measurements must be taken to confirm their proper alignment with respect to the fixed axes of the simulator. Often the control stick is required to be rotated several degrees off axis. A protractor can be used to verify the correct angle.

Calibration of the control stick is similar to that of any transducer. A force is applied to the stick and the resulting output voltage is recorded. The force vector must be along the true axis of control determined previously. Care must be taken to apply the force at the same point on the stick every time it is calibrated to insure consistent results. This point should be at the nominal hand grip centerpoint and through the center of the stick. Data must be collected at regular intervals throughout the active range of the device. This procedure must then be repeated for the opposite stick axis.

Once this has been completed, the data can be plotted for visual inspection. A first order least squares fit is often helpful to determine the slope and intercept of the line. The fit should be linear, symmetrical, and have a small offset. Offsets reduce the symmetrical dynamic range of the device. Offsets can be calculated by the simulation software before each run and corrected for in real-time if they are small. Hardware problems should be suspected and investigated if the calibration procedure uncovers any sizable deviations from these characteristics.

The raw stick command voltages are sampled under software control and are usually conditioned by imposing a breakout force, scaling and limiting. The breakout force is an artificially introduced control response deadband at and near zero as depicted by ± Min Input in figure 9. The deadband is useful for eliminating small, unintentional inputs (e.g. from the hand resting on the stick) and sampling noise. Limiting sets the absolute maximum and minimum allowable control input and is shown as ± Max Output in figure 9. These critical points can be easily verified by comparison of time history data from the raw and conditioned stick signal from a simulation run with full-range control inputs.

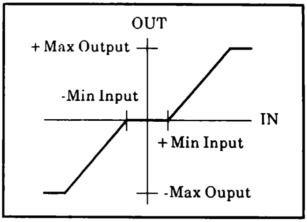


Figure 9. Stick Signal Conditioning

A method used to ensure maximum signal-to-noise ratio on the control stick channel is automatic gain switching. This entails simultaneously boosting the command voltage with three or more amplifiers of different gains. The software samples all of these signals at the same time. The signal with the highest gain that is not saturated is scaled in accordance with its amplification and input to the simulation. The gain of each amplifier must be determined accurately and checked frequently. This can be accomplished easily using a frequency response analyzer. The gain circuits must exhibit a linear phase so no time distortions are introduced.

Engine throttles in flight simulators are typically potentiometers adjusted by the throttle lever. Data must be collected on output voltage between the forward and rear lever limits to allow the correct scale factor to be determined. The engineer/technician must also verify that the throttle limits are reached before the analog-to-digital converter saturates.

Flight simulations usually include visual displays which are directly or indirectly affected by control input scaling. For example, a fuel flow meter display is a direct function of throttle setting and must be calibrated to indicate the proper flow for any given throttle lever position. Roll and pitch commands from the stick are typically scaled in degrees per second per pound-force. By applying a constant stick force and measuring the time ($T_{360°}$) to complete one revolution, the actual control scale factor can be computed using equation 9. This can then be verified by comparison with the theoretical scaling.

Control Scale Factor (°/sec/lb) $= 360° \div T_{360°}$ (sec) ÷ Force (lbs) (9)

This timing check provides an excellent end-to-end verification as well because it also includes the aircraft dynamics, simulation to graphics computer communications and graphics equations.

Numerical/Mathematical Considerations

Numerical integration of any vehicle state must be verified to make sure the integration technique is appropriate, working properly, and is being initialized and re-initialized properly. In simulations using a high sample rate, simple trapezoidal integration is most likely appropriate. However, in other situations a more sophisticated technique may be required. After selecting the best technique for a particular situation, the operation of the integration software must be verified to make sure that the code is indeed integrating. Visual inspection of the time history data is not enough. Two other verification procedures are suggested. First, if a known function is input to the integrator, then the result can be compared to the analytical solution. Second, if the simulation is using a sum-of-sines as a forcing function and there is the capability for frequency domain analysis, then a transfer function can be computed for the input and output of the integrator to verify that it is working properly. The transfer function should have a slope of -20 dB per decade for the gain and a constant -90 degrees for phase. It is also important to verify that the integrator is being initialized and re-initialized properly. The initial condition of the integrator must be set equal to the exact starting value of the the vehicle. Also, if the simulation is being reset or re-initialized, then it is important to reset the initial conditions of integrators and to re-initialized the variable(s) that holds the past value(s) used in the integration equation(s).

The Euler transformation matrix and equations of motion should also be verified to make sure there are no terms missing. In research simulators where the task requires small perturbations about the initial vehicle state, it is possible to have missing terms go unnoticed. However, the missing terms can effect the collected data and thus the reporting of the results.

Verify the Scoring Algorithm

Simulators that are used in research typically incorporate a scoring algorithm to measure pilot performance. Verification of the scoring algorithm is very important Pre-calculated patterns may be injected into the simulation to verify that the expected score is calculated. It is also important to verify that the scoring starts and stops at the proper times. The code should also be checked to verify that all temporary variables used in the scoring algorithm are being re-initialized properly between trials.

Conclusion

Although these procedures evolved in a research laboratory, their applicability is not solely limited to research simulators. Some of these procedures apply to all real-time simulators, even beyond aircraft simulators. The procedures discussed in this paper are mostly quantitative, however, subjective evaluation is also encouraged. Specifically, visual inspection of collected time history data, in the form of plots, can often reveal peculiaralities. Secondly, subjective comments from pilots can often lead to valuable investigation. Knowing that these set of procedures has been rigorously followed gives the researchers a high degree of confidence in the collected data.

References

1. Levison, W.H. (May 1982). Methods for Identifying Pilot Dynamics. Procedings of the Workshop on Flight Testing to Identify Pilot Workload and Pilot Dynamics. AFFTC-TR-82-5.

2. Oppenheim & Schafer (1975). Digital Signal Processing, Prentice-Hall.

3. Gum, D.R. & Martin, E.A. (1987). The Flight Simulator Delay Problem. AIAA paper 87-2369-CP, Montery, CA.

4. Military Specification (1980). Flying Qualities of Piloted Airplanes. MIL-F-8785C, ASD/ENESS, Wright-Patterson AFB, Dayton, Ohio.

5. Riccio, G.E., Cress, J.D. & Johnson, W.V. (1987). The Effects of Simulator Delays on the Acquisition of Flight Control Skills: Control of Heading and Altitude. Procedings of the 31st Meeting of the Human Factors Society. New York, NY:Human Factors Society.

6. Johnson, W.V. & Middendorf, M.S. (1988). Simulator Transport Delay Measurement using Steady State Techniques. AIAA paper 88-4619-CP, Atlanta, GA.

7. Levison, W.H. (1975). Techniques for Data Analysis and Input Waveform Generation for Manual Control Research. Technical Memorandum, CSD-75-2, Bolt, Beranek, and Newman. Cambridge, MA.

89-3261-CP

DEVELOPMENT AND VALIDATION OF AN F-16C AERODYNAMIC MODEL FROM FLIGHT TEST DATA

Daniel P. Ringenbach, 1Lt., USAF
Edwards Air Force Base, California

Abstract

A realtime aerodynamic model for the F-16C simulation was developed solely from data acquired from flight test. The model includes major aerodynamic derivatives in pitching, rolling and yawing moments as well as derivatives in normal, chord and side forces. Pilot inputs of lateral and longitudinal stick force, rudder pedal force, power lever angle and speedbrake deflection recorded during flight test were used as input into the simulator and the responses of the aircraft and simulator were compared. The study showed that an aerodynamic model from flight test data showed a substantial improvement over the aerodynamic model developed from wind tunnel data in the high angle-of-attack region but was not extensive enough to develop a simulator for the entire flight envelope. It also showed that the aerodynamic model developed from wind tunnel data was inadequate and should be updated with data acquired during flight test.

Introduction

The Test and Evaluation Mission Simulator (TEMS) complex at the Air Force Flight Test Center (AFFTC) develops flying qualities simulators for aircraft currently being tested and aircraft soon to be tested at Edwards Air Force Base. Typically an aircraft-specific aerodynamic model is developed for the simulator using contractor supplied data from wind tunnel tests. This data is used along with standard equations of motion, atmospheric properties and an aircraft-specific flight control system model to create a realtime simulation of the aircraft. The TEMS complex provides flying qualities and flight control system test engineers with an efficient and inexpensive way to plan flight test programs, perform sensitivity studies, design test maneuvers, plan and rehearse flight test missions, prefly specific test points, investigate anomalies and evaluate pilot-in-the-loop dynamics.

The F-16 Combined Test Force had a requirement to use the simulator to evaluate and possibly predict departure characteristics of the F-16C. The standard F-16 simulation was not sufficiently accurate in the area of the envelope of interest (high angle-of-attack region). Rather than adjusting the wind tunnel aerodynamic model, the 6516 Test Squadron and TEMS engineers developed an aerodynamic model of the F-16C from November 1988 through March 1989 using data acquired during flight test. With a simple classical linear model made up of the basic aerodynamic derivatives, it could be readily validated against actual flight test results. And with a validated model, the simulator could be used to prefly hazardous test points, reduce actual test flying time, assist with the handling qualities evaluation, evaluate and design potential changes to the flight control computer to remedy anomolies discovered in flight test and possibly predict a departure.

Method of Development

Data Extraction

The first stage in developing the aerodynamic model was already accomplished during flight test in April 1988. Data recorded during pitch, yaw and roll doublets performed in flight were used to extract aerodynamic derivatives using the Modified Maximum Likelihood Estimator computer program (MMLE) on the AFFTC CDC Cyber computers.

Two sets of data were generated with MMLE so that flexibility effects were separated from rigid body behavior. The rigid body data set had aerodynamic derivatives as a function of angle-of-attack (0 to 28 degrees) and Mach number (0.6, 0.8, and 0.9). The flexibility effects data had derivatives as a function of Mach number (0 to 2.0) and dynamic pressure (200, 400, 800, and 1600 lb/ft^2). Unfortunately, there was a significant scatter in the data for all aerodynamic derivative which made it difficult to establish definitive curves. By studying derivatives in test reports and wind tunnel data on previous F-16′s , the general shape of the curves were drawn through the points. These fairings were made for three store loadings of the F-16C aircraft and were used to build an aerodynamic model for the simulator.

Simulation Database Formulation

Data Input

The fairings of the plots generated by MMLE were entered into the TEMS complex development computer using a digitizer. Each plot became a two dimensional table in the F-16 simulation aerodynamic derivatives database and was formatted to make it compatible with the present F-16 simulator.

The flexibility effects data were valid for low angles-of-attack (0 to 4 degrees). The rigid body data were valid for the angles-of-attack in the range of interest but only at dynamic pressures below 200 lb/ft^2. The challenge was to implement the two sets of data and correlate them in such a way as to get valid aerodynamic derivatives over the required range of interest of angle-of-attack, Mach number and dynamic pressure.

Rigid Aerodynamic Data and Flexibility Effects

The two sets of data had crossplot points which were used to correlate the data. Using these points, a series of steps combined the two plots for each derivative into one which was dependent on angle-of-attack, Mach number and dynamic pressure. The crossplot points were valid for level unaccelerated flight at a dynamic pressure below 200 lb/ft^2 (Fig. 1).

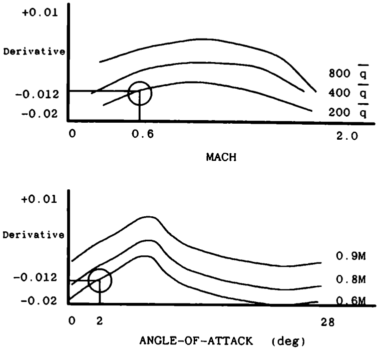


Fig. 1 Illustration of Crossplot Points

Once the crossplot points were determined, the flexibility effects for each Mach number were calculated. Since the rigid body data had only three lines of Mach number, it was only necessary to calculate flexibility effects at 0.6, 0.8, and 0.9 Mach numbers. In the flexibility effects data, the value of each specific derivative at a specific Mach number and separate dynamic pressure was measured. The difference between the two values was then divided by the difference in dynamic pressure to produce the flexibility correction for each derivative. The flexibility correction was assumed to be constant at each Mach number. Equation (1) was used to compute the flexibility correction for each derivative.

$$\text{Flexibility Correction } \bar{q}_{cor} = \frac{\Delta \text{ Derivative}}{\Delta \bar{q}} \quad (1)$$

The next step was to define the zero dynamic pressure point for each coefficient. The zero points were calculated using the following equation.

$$\text{zero point} = -(\bar{q}_{cor})(200) + (\substack{\text{derivative}\\\text{value}} @ 200 \ \bar{q}) \quad (2)$$

Once the zero point was found, the rigid body data was corrected to zero dynamic pressure by adjusting the curve to go through the zero point. By measuring the difference between the crossplot point and value of the curve for a given angle-of-attack and adding that difference to the zero point value, a new curve was generated (Fig. 2).

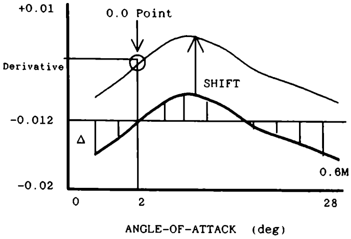


Fig. 2 Generation of Zero Dynamic Pressure Curve

This new data was the baseline for the flight test aerodynamic model. With the zero dynamic pressure data, the derivatives were dependent only on angle-of-attack and Mach number. The effect of dynamic pressure was added directly to the value of the derivative obtained from the zero dynamic pressure data using the equation below.

$$\text{Corrected Derivative} = \substack{\text{derivative}\\\text{value}} @ 0 \ \bar{q} + (\bar{q}_{cor})(\bar{q}) \quad (3)$$

Table Look-up Generator

The zero dynamic pressure data tables, along with tables of flexibility corrections for each derivative, were input into a database editor program (DBEDIT) at the TEMS complex. DBEDIT generated a FORTRAN blockdata subroutine which contained all derivative and flexibility correction tables and a FORTRAN look-up subroutine which used linear interpolation to calculate derivative values. The computer generated code was ready for compilation and linking.

Total Coefficient Equations

A separate subroutine was used to calculate the flexibility corrections and the total coefficient equations. This subroutine took the outputs of the look-up subroutine and calculates the new coefficient value using equation (3). Once this was done, the total coefficient equations (4) through (9) were solved and the output from this subroutine was fed into the equations of motion subroutines.

PITCHING MOMENT

$$Cm = Cm_o + Cm_{\alpha} \cdot \alpha + Cm_{\delta h} \cdot \delta h + Cm_q \cdot Q \cdot \bar{C}/2V \qquad (4)$$

ROLLING MOMENT

$$Cl = Cl_{\beta} * \beta + Cl_r * R * b/2V + Cl_p * P * b/2V + Cl_{\delta r} * \delta r + Cl_{\delta a} * \delta a \qquad (5)$$

YAWING MOMENT

$$Cn = Cn_{\beta} \cdot \beta + Cn_r \cdot R * b/2v + Cn_p * P * b/2V + Cn_{\delta r} \cdot \delta r + Cn_{\delta a} \cdot \delta a \qquad (6)$$

NORMAL FORCE

$$C_{NF} = C_{No} + C_{N\alpha} \cdot \alpha + Cm_{\delta h} \cdot \delta h \cdot \text{Multiplier} \qquad (7)$$

SIDE FORCE

$$C_{YF} = C_{Y\beta} \cdot \beta + C_{Y\delta a} \cdot \delta a + C_{Y\delta r} \cdot \delta r \qquad (8)$$

CHORD FORCE

$$C_{CF} = C_{Co} + C_{C\alpha} \cdot \alpha \qquad (9)$$

Simulation Interface

The flight test aerodynamic model was built to work with the current F-16 simulation. The TEMS complex has a realtime computing system consisting of four Perkin-Elmer 8/32 computers communicating via shared memory. The F-16 simulator uses three of these CPUs along with an AYDIN line generator for a visual display and a fixed base F-16 cockpit. Figure 3 shows the software set up of the F-16 simulator. The master CPU executes the flight control system software and I/O subroutines for the cockpit, visual system and realtime disk. The first slave CPU executes the flight test aerodynamic model and the equations of motion. The second slave executes the engine model, automatic trim, weight and balance calculations and a flight control system gain setting subroutine. All of these subroutines are sequenced and run in a frame time of 15.63 milliseconds (64hz).

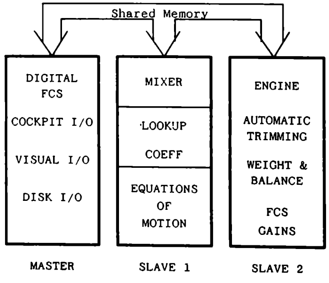


Fig. 3 F-16 Simulation Software Set Up

Simulator Verification

A number of points within the envelope of the flight test database were chosen by the TEMS engineer to check out the table look-ups and coefficient calculations. For all the points chosen, the simulator value of the derivative fell within five percent of the predicted value which was taken from the original MMLE data. To fully verify that the table look-ups were correct, an engineer from the 6516 Test Squadron conducted a separate set of static checks and the results were similar. Differences between the simulation value and predicted value can be attributed to errors in digitizing, rounding errors in flexibility effect calculations, and rounding errors in total coefficient calculations.

Validation With Flight Test Data

For the simulator to be truly useful it had to be validated to show that the model could recreate actual aircraft response. With sufficient validation data, the model could be used to collect data and conclusions could be made about the aircraft from this simulator data.

The simulator validation effort focused at first on using test pilots to fly the simulator. They flew the same maneuvers in the simulator as were flown in the aircraft during flight test. The data were displayed on strip chart recorders and then compared to the data recorded during flight test missions. A number of problems hampered the effort. First, test pilots were available on a limited basis and were not able to spend the time in the simulator necessary to collect sufficient data to validate the simulator. Secondly, although the pilot inputs were similar, they were not exactly the same. This made comparisons between the simulator and aircraft responses difficult. And lastly, the strip chart recorder scales were large to account for the minimum and maximum values of the recorded parameters and gave no indication of the fine differences between simulator and aircraft. These problems made this method of simulator validation unsatisfactory.

Flight Test Inputs

The solution to validating the simulator and streamlining the process was to use pilot inputs recorded during flight test as simulator inputs. This solution eliminated the need for a test pilot to fly the simulator and also provided the exact inputs the pilot used during the in-flight maneuver. The simulator and aircraft responses could be compared directly providing that care was taken to set up the simulator with the same initial conditions as the aircraft. If the responses were identical, the simulator was considered valid for that flight condition. If they were not identical, changes were made to aerodynamic model to improve the simulator.

Using prerecorded pilot inputs in the simulator was not an easy task. It required the flight test inputs be read from a disk in realtime. A program was written and introduced into the master CPU which could read up to ten parameters. The 6516 Test Squadron engineers recorded a number of maneuvers such as constant heading sideslips and 360 degree rolls onto magnetic tape. Each maneuver was stored as a separate file on the realtime disk which was accessed by the simulation engineer during simulator operation. When accessed, stick force, rudder pedal force, power lever angle and speedbrake deflection were input into the simulator as a pilot would if he were using the cockpit. The maneuver could be repeated as many times as needed. Figure 4 shows how the cockpit was replaced by the disk.

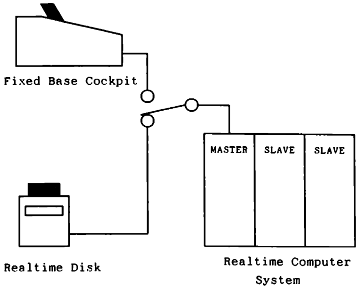


Figure 4: Flight Test Input Set Up

Simulator and Aircraft Comparisons

A number of different maneuvers from several test sorties were stored on the real time disk. Using the disk as input these sorties were flown in the simulator. Angle-of-attack, sideslip angle, roll angle and pitch, roll and yaw rates were recorded using strip chart recorders and then compared to the flight test strip chart recordings. At first there were very notable differences between the simulator and aircraft responses. For example, on a constant heading sideslip, there was only a few degrees of roll angle for the aircraft but the simulator roll angle was nearly 180 degrees. This pointed out a problem with roll derivatives such as $C_{l\beta}$ and $C_{l\delta a}$. By changing these in realtime and running the same maneuver again, the effects of the change were recorded and improvements to the aerodynamic model were made.

In order to change the derivatives in realtime a multiplier and bias were provided for each derivative. These could be changed by the simulation engineer very quickly on the simulator control terminal. By noting the changes to the derivatives, the baseline data (MMLE) were replaced by new fairings. Once the new fairings were in place, the multipliers were reset to one and biases to zero. Several iterations of this process have been completed.

Once a good approximation was observed on the strip charts, the next step was to look closely at the aircraft response and simulator response. To do this, the simulation engineer used software already available on the real time system to record a number of parameters during the maneuver on the realtime disk. This data and the aircraft data were then loaded into a VAX Station 2000 Workstation. Using MATRIX-X, an analysis and graphics software package, the aircraft and simulator responses were plotted and overlaid on the same page. The scale was enlarged to clearly show the differences between the simulator and the aircraft. In addition, the simulator responses using the Wind Tunnel Model was also overlaid.

Results

All of the results presented are from one test sortie in which constant heading sideslips and 360 degree rolls were performed with angles-of-attack from 12 degrees to the 26 degrees (the F-16 flight control system has an angle-of-attack limiter set at 26 degrees). The following parameters were recorded during the flight using AFFTC range facilities. The same parameters were recorded from the simulator at the TEMS complex.

Angle-of-Attack	(deg)
Sideslip angle	(deg)
Flaperon deflection	(deg)
Rudder Deflection	(deg)
Horizontal Tail Deflection	(deg)
Roll Rate,Body Axis	(deg/sec)
Pitch Rate,Body Axis	(deg/sec)
Yaw Rate,Body Axis	(deg/sec)
Roll Angle	(deg)
Normal Acceleration	(g's)
Calibrated Airspeed	(knots)
Altitude	(feet)
Lateral Stick Force	(lbs)
Longitudinal Stick Force	(lbs)
Rudder Pedal Force	(lbs)

The first comparison made was between flight test data and the F-16 simulator with the aerodynamic derivatives supplied from wind tunnel tests (Wind Tunnel Model). This model had aerodynamic derivative increments for store loadings and derivative increments for angles-of-attack up to 40 degrees. Care was taken to ensure that the proper initial conditions were set before using the flight test data input into the simulator. Once the conditions were set, the simulator first flew a constant heading sideslip at 12 degrees angle-of-attack. Figure 5 shows the comparisons between aircraft response and the Wind Tunnel Model response to the same flight test inputs. In this case the aircraft had maximum and minimum values of sideslip angle of 12 and -13 degrees while the Wind Tunnel Model had maximum and minimum values of 12.5 and -12.5 degrees. Figure 6 shows the same maneuver at 20 degrees angle-of-attack. The aircraft had maximum and minimum values of sideslip angle of 7.0 and -9.0 degrees while the Wind Tunnel Model had maximum and minimum values of 14 and -17.5 degrees. The data shown on these plots demonstrated that at high angles-of-attack the Wind Tunnel Model failed to closely model the aircraft response in sideslip angle.

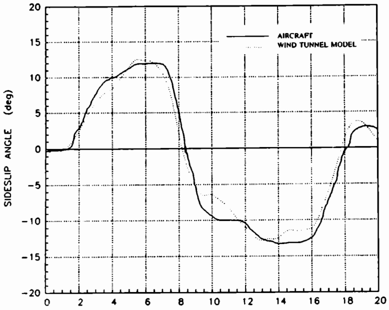


Fig. 5 12deg AOA CONSTANT HEADING SIDESLIP

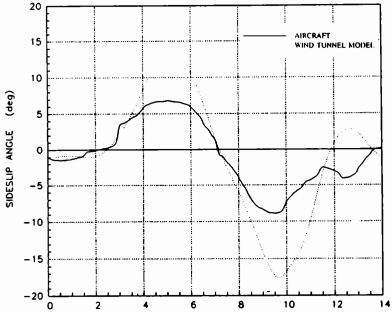


Fig. 6 20deg AOA CONSTANT HEADING SIDESLIP

The model developed from flight test data (Flight Test Model) was also compared to the aircraft response and the Wind Tunnel Model response. Figure 7 shows responses of both models and the aircraft for the constant heading sideslip at 20 degrees angle-of-attack. The Flight Test Model had maximum and minimum values of sideslip of 7.5 and -10.5 degrees which closely approximated the aircraft response. Similarly, in another flight test maneuver, the Flight Test Model approximated the aircraft response more closely than the Wind Tunnel Model. Figure 8 shows the amount of sideslip angle generated in a 360 degree left-hand roll at an angle-of-attack of 22 degrees. The aircraft

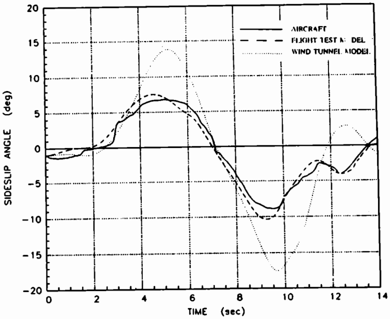


Fig. 7 20deg AOA CONSTANT HEADING SIDESLIP

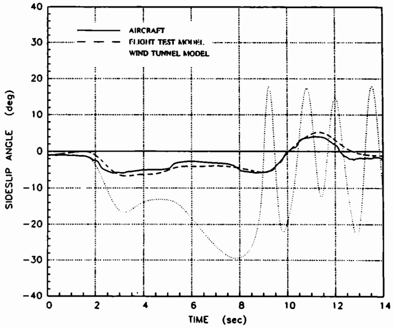


Fig. 8 22deg AOA 360deg LEFT ROLL

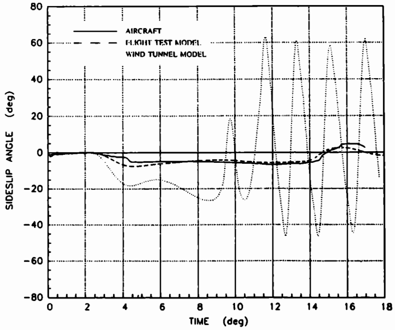


Fig. 9 26deg AOA 360deg LEFT ROLL

had a maximum value of sideslip angle of about 6 degrees while the Flight Test Model showed about 7 degrees of sideslip angle. The Wind Tunnel Model had a large sideslip angle excursions of nearly 30 degrees and departed from normal flight before the maneuver was complete. A similar result also occurred for a 360 degree left-hand roll at the angle-of-attack limiter of 26 degrees. The aircraft had a maximum sideslip angle of about 6 degrees and the Flight Test Model had a maximum of 7 degrees but held about 6 degrees of sideslip angle throughout the maneuver. The Wind Tunnel Model showed a sideslip angle of about 25 degrees before departing from normal flight.

Conclusions

Several conclusions can be made from this study. First it is necessary to validate a simulation against the best available data prior to using it. As was seen with the Wind Tunnel Aerodynamic Model, the simulator and aircraft had very different responses at high angles-of-attack. A second conclusion is that while using flight test data to build a simplified model has proven to be accurate in some cases, there is definitely not enough data available to develop a full simulation from flight test data alone. The simple linear model using MMLE generated data may not take into account all of the possible stability derivatives and may never model the non-linearities of the aircraft correctly for all cases. And a final conclusion is that data acquired during flight test can be used to validate and update the simulator aerodynamic model. By using the VAX Station 2000 and the Matrix-X software to compare the aircraft and simulator responses, the simulator can be updated with flight test data and validated quickly and efficiently. This method will be applied to other simulators at the TEMS complex which will increase its utility to the flight test programs and its ability to support and augment flight test.

89-3262-CP

USE OF A SIMPLIFIED ESTIMATION SCHEME FOR SIMULATION VALIDATION AND IMPROVEMENT

Robert A. Hess* and Peter L. Ly**

Flight Simulation Group
Systems Control Technology, Inc.
100 Exploration, Suite 2005
Lexington Park, MD 20653

ABSTRACT

This paper discusses the development and use of an estimation scheme imbedded into a simulation environment. This scheme allows the simulation engineer to model errors in the simulation response as intrinsically linear functions of simulation variables by comparing the simulation response to response taken from actual systems. The applicability of this tool is demonstrated.

NOMENCLATURE

Symbol	Description
C	dependent variable
e	error vector, length N
F_{P_i}	i^{th} partial F-statistic
K_i	i^{th} parameter value
L	least-square estimator performance measure
m	number of parameters in model (excluding constant term)
N	number of samples of data
P	error parameter vector, length m (or m+1)
P_i	i^{th} error parameter
R^2	squared multiple correlation coefficient
SS_E	error sum of squares
SS_R	regression sum of squares
S_Y^2	variance of fit estimate
S_{YY}	total sum of squares
X_i	value of i^{th} independent variable at specific time
X_{ij}	i-j element of matrix X
Y	dependent variable, length N
σ_{P_i}	standard deviation of i^{th} parameter estimate
σ_Y^2	standard deviation of fit
$\mathcal{X}$	independent variable matrix, size m x N

superscripts

^	denotes estimate of value
T	denotes matrix transpose
-1	denotes matrix inversion
_	denotes mean of signal

INTRODUCTION

Systems Control Technology, Inc. (SCT) has previously developed a tool for simulation validation. This tool, known as SCOPE (Simulation Checking using an Optimal Prediction Evaluation), allowed the simulation engineer/scientist the ability to exercise various portions of a simulation independent of one another in an effort to evaluate each portions fidelity (Reference 1). Hence, the simulation specialist obtained the ability to ascertain the performance of the aerodynamic, control system, and engine models (among others) by validating such models against test data collected from other simulations or collected in flight.

SCOPE was designed to allow the simulation specialist to treat each portion of the simulation in an 'open-loop' fashion. Each input to the model can be read into the simulation and used to 'drive' the simulation module. The response of the module is collected and saved along with the inputs to disk files for additional processing. In the case of control system validation, it is usually evident what kind of modeling error occurs in the control system model. This is because structure of the control system model is well known, and implementation errors in the control system model are easily traceable. This is not usually the case in aerodynamic model validation. In general, the actual aerodynamics of an aircraft are unknown. Furthermore, there are many independent inputs which can affect the outputs of the aerodynamic model. Hence, when performing and aerodynamic model evaluation, it can be very difficult to determine why the model performed poorly.

LOST (Low Order eStimation Tool) resulted from the need to allow the simulation specialist the ability to pinpoint the exact problem in multi-input, linear/intrinsically linear models (such as aerodynamic models). LOST compares the model's response to measured inputs to measured response. Systematic errors in the model are correlated with various user-specified inputs, and corrections to the model are calculated. LOST also provides statistics about the corrections which can aid the user in judging the importance of each correction. With these corrections, the bewildering problem of realizing a solution to the simulation modeling problem is made easier.

THEORETICAL BACKGROUND

Overview to Simulation Modeling

An aircraft simulation is composed of a variety of computer software components or modules, each of which is implemented in a serial fashion (i.e., module 1 provides inputs to module 2, which provides inputs to module 3, etc.)

Each of these modules is a mathematical model of some physical aspect of the actual system and has known inputs and known outputs. For example, the aerodynamic forces and moments of an aircraft are modeled in an aerodynamic module, whereas the control system response is modeled in a control system module. In the case of aerodynamic modules, the mathematical model is one which is multi-input and intrinsically linear. In other words, the model is generally composed as a component build-up of many terms, each a function of a separate input of the form:

$$\hat{C} = K_0 + C_1(X_1) + C_2(X_2) + C_3(X_3) + \ldots \qquad (1)$$

* Engineer, member AIAA
** Engineer

where $\hat{C}$ is the response of the model (such as a lift coefficient) and C_i are the incremental effects due to the various inputs, X_i. The inputs, X_i, are composed of combinations of various state and control signal, such as α, β, α^2, δA, $\alpha\ \delta A$, etc. Though these signals may be non-linear combinations of state and control signals, the resulting combination can be considered a new signal.

In the case of aerodynamic modeling, each of the incremental effects can be mathematically modeled as a linear combination of parameters and inputs of the form:

$$C_i = K_i X_i \qquad (2)$$

Each of the K_i parameters effectively represent the aircraft aerodynamic stability and control characteristics. Physically, such parameters have relationship to one another in explaining the flying qualities and performance of an aircraft. Such a representation of an aircraft's aerodynamics provides an equivalent system model of the actual aerodynamic phenomena.

Error modeling

The mathematical model incorporated in a simulation aerodynamic module is only a mathematical construct of an actual system. Such a model may not accurately explain the actual aerodynamic forces and moments adequately. This can be due to the following:

a) the parameter values, K_i, may be in error
b) the specified inputs, X_i, may be incomplete

Assume that the actual aerodynamic force/moment coefficient is measured**, and represented by the variable C. This measured response is created due to changes in control surface position, changes in flow direction and velocity, and due to aircraft dynamic maneuvering. The difference between the measured and modeled coefficients is given by

$$Y = C - \hat{C} \qquad (3)$$

Due to the nature of the aerodynamic model in the simulation, it is of interest to model this error in a manner similar in structure to the mathematical representation incorporated in the simulation model. Hence, the error can be modeled as:

$$Y = P_0 + P_1 X_1 + P_2 X_2 + \ldots \qquad (4)$$

Each of the P_i parameters can be thought of as increments to the K_i parameters needed to match the measured and modeled force/moment coefficients.

Error Parameter Estimation

For the sake of argument, let us denote Y as the dependent variable, X_i as the independent variables, and P_i as unknown parameters. As we are dealing with data created/stored in a digital format, we have a discrete collection of samples of data, each sample providing a one-to-one correspondence between the dependent and independent variables:

$$\begin{array}{l}
Y_1 \sim X_{11}\ X_{12}\ X_{13}\ X_{14}\ \ldots\ X_{1m} \\
Y_2 \sim X_{21}\ X_{22}\ X_{23}\ X_{24}\ \ldots\ X_{2m} \\
\\
Y_N \sim X_{N1}\ X_{N2}\ X_{N3}\ X_{N4}\ \ldots\ X_{Nm}
\end{array} \qquad (5)$$

where m is the number of independent variables, and N is the number of observations of Y.

The relationship between the dependent and independent variables can be written as a series of independent equations:

$$\begin{array}{l}
Y_1 = P_0 + P_1 X_{11} + P_2 X_{12} + \ldots + P_m X_{1m} \\
\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdots \quad \cdot \\
Y_N = P_0 + P_1 X_{N1} + P_2 X_{N2} + \ldots + P_m X_{Nm}
\end{array} \qquad (6)$$

or in matrix form as:

$$Y = P\mathcal{X} \qquad (7)$$

where $\mathcal{X}$ is the (m+1) x N matrix of the form:

$$\mathcal{X} = \begin{array}{lll}
1 & X_{11} & X_{12}\ \ldots \\
1 & X_{21} & X_{22}\ \ldots \\
1 & X_{31} & \cdot\ \ \ldots \\
1 & X_{41} & \cdot\ \ \ldots \\
1 & X_{N1} & \cdot\ \ \ldots
\end{array} \qquad (8)$$

and where Y is the vector composed of each measurement of the dependent variables (length N) and P is the m+1 vector of unknown parameters.

It is of interest to estimate the value of P such that the resulting estimated Y vector contains as little squared error as compared to the actual Y vector. If we represent the error between the actual and estimated dependent variable as

$$e_i = Y_i - \hat{Y}_i \qquad (9)$$

then the sum of squared error is given by the performance measure of the form:

$$L = \sum_{i=1}^{N} e_i^2$$

$$= e^T e = Y^T Y - 2P^T \mathcal{X}^T Y + P^T \mathcal{X}^T \mathcal{X} P \qquad (10)$$

It is of interest to determine the parameter values which minimize this error. Hence the parameters must satisfy the relationship

** Actually, aerodynamic force/moment coefficients are not directly measurable, but must be reconstructed from measured signals. This reconstruction is generally an algebraic manipulation, and hence, fairly straightforward.

$$dL/dP|_P = -2\mathcal{X}^T Y + 2\mathcal{X}^T \mathcal{X} P \quad (11)$$

which can be solved for the parameter estimates $(\hat{P})$ to yield:

$$\hat{P} = (\mathcal{X}^T \mathcal{X})^{-1} \mathcal{X}^T Y \quad (12)$$

This is the solution of the classic multiple linear regression problem (Ref. 2).

Calculation of Error Parameter Statistics

The equation for the estimates of the error parameters presented in the previous section provides only a partial understanding as to the adequacy of the estimated parameters. It is of additional importance to determine the statistical significance of each parameter. This includes the evaluation of each parameter's confidence bound, their significance in the model, and their combined effect in explaining the dependent variable.

An estimate of the dependent variable is formulated from the estimated parameters and the independent variables from the equation

$$\hat{Y} = \hat{P} \mathcal{X} \quad (13)$$

The mean (average) value of the dependent variable is given by

$$\bar{Y} = \frac{1}{N} \sum_{i=1}^{N} Y_i \quad (14)$$

This approximation approaches the true sample mean as the number of samples approaches infinity.

A measure of the amount of scatter in the estimated dependent variable as compared to the actual dependent variable is a statistic known as the variance of fit. It is given by the relation:

$$\sigma_Y^2 \cong S_Y^2 = \frac{1}{N-m-1} \sum_{i=1}^{N} e_i^2 \quad (15)$$

Note that S_Y^2 is only an approximation to σ_Y^2 since we are dealing with a finite amount of sampled data.

One of the most common statistics used when fitting a model to data is the percent fit to data criteria, R^2 (also known as the squared multiple correlation coefficient and/or the coefficient of determination). It is a measure of the ability of the estimated model/parameters to fit the data. R^2 is defined as:

$$R^2 = SS_R / S_{YY} = 1 - [\, SS_E / S_{YY}] \quad (16)$$

where

$$SS_R = \sum_{i=1}^{N} (\hat{Y}_i - \bar{Y}_i)^2 \quad (16a)$$

$$SS_E = \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2 \quad (16b)$$

$$S_{YY} = \sum_{i=1}^{N} (Y_i - \bar{Y}_i)^2 \quad (16c)$$

Each of the determined parameter values is only an estimate of the actual parameter value. Hence, it is important to determine the confidence bounds of the estimated parameters. The square-root of the variance on each parameter is given by:

$$\sigma_{P_i} = i^{th} \text{ element of } \sqrt{\{S_Y^2 \ \mathrm{diag}[(\mathcal{X}^T \mathcal{X})^{-1}]\}} \quad (17)$$

By definition, the confidence bound of each parameter are represented by a value equal to 2 times the σ_{P_i} value. Large values of σ_{P_i} (relative to the parameter value) indicate that there is uncertainty in the estimate of P_i. It is desirable to have the confidence bounds on each parameter as small as possible. Note, the development of σ_{P_i} assumes that each of the independent variables are truly independent and uncorrelated. If so, then the matrix $(\mathcal{X}^T \mathcal{X})^{-1}$ will be a diagonal matrix. In actuality, the independent variables are usually not truly independent and uncorrelated. This means that $(\mathcal{X}^T \mathcal{X})^{-1}$ will be a non-diagonal matrix, with the off-diagonal j-k elements representing the levels of correlation between variables j and k. If the independent variables are not independent, then the confidence bounds determined using this formulation can be misleading.

It is useful to define a measure of statistical significance for each of the estimated parameters. This is performed so that unobservable/insignificant terms can be identified and removed from the model. One measure of parameter significance is the partial F-statistic, defined as

$$F_{P_i} = P_i^2 / \sigma_{P_i}^2 \quad (18)$$

A parameter is considered significant if F_{P_i} is greater than some critical value, F_{crit}. F_{crit} is based on the F-distribution $F(n_1, n_2, \alpha_P)$, where $n_1 = 1$, $n_2 = N-m$, and having a significance level α_P**.

** For aerodynamic identification problems, F_{crit} usually can take on values between 5 and 20. A partial F-statistic greater than 20 generally means that the estimated parameter is significant, and should be left in the model.

An F_{crit} value less than 5 usually indicates that the estimated parameter is insignificant, and need not be incorporated into the model. Values of F_{P_i} between 5 and 20 indicate a parameter has marginal acceptability in the model.

IMPLEMENTATION

The LOST software is composed of a collection of Pascal modules which interface directly with a simulation environment and the SCOPE software. LOST consists of modules to build all the neccessary vectors needed to estimate increments to the system parameters. Furthermore, LOST contains all the routines needed to provide the user with statistics on the estimates, which gives the user the ability to check parameter significance and model adequacy.

Illustrated in Figure 1 is a data flow diagram indicating the flow of information throughout the LOST software.

LOST has been designed to be used in an interactive computer session. This allows the simulation engineer the ability to experiment with a variety of model structures very quickly during a simulation session.

Once a LOST analysis has been chosen, the screen will display the LOST menu. This menu (shown in Figure 2) contains 4 items to be performed:

1) choice of dependent variable
2) choice of independent variable(s)
3) execute least-squares analysis
4) return to SCOPE menu

To perform a LOST analysis, the user must specify the independent and dependent variables, and execute the LOST routines. The user is able to change, save and retrieve dependent and independent variables and perform LOST analyses as often as wanted before exiting this menu.

When the LOST analysis is completed, the results are displayed to the screen as shown in Figure 3. The same results are also written to a disk file.

A variety of information is displayed to the screen:

1) the dependent variable
2) the R^2 criteria
3) variance of fit to data
4) a list of independent variables
5) the values of the parameters
6) the confidence bounds associated with each parameter
7) the partial F-statistic for each estimate

For example, the display in Figure 3 indicates that the error in the rolling moment coefficient (CLL) was fit to 91.35%. The variance of the fit was 4.66891×10^{-4}. The model for the error in the rolling moment aerodynamic database should be of the form

$$\begin{aligned}\text{error in CLL} = & - 6.839\text{x}10\text{-}4 \; - \; 5.184\text{x}10^{-4}\,\text{BETA} \\ & + 2.655\text{x}10^{-2}\,\text{BETA}^2 \; - \; 8.254\text{x}10^{-2}\,\text{PB} \\ & + 3.618\text{x}10^{-3}\,\text{PB ALFA}\end{aligned}$$

where ALFA, BETA, PB and CLL are the variable names used in the actual simulation.

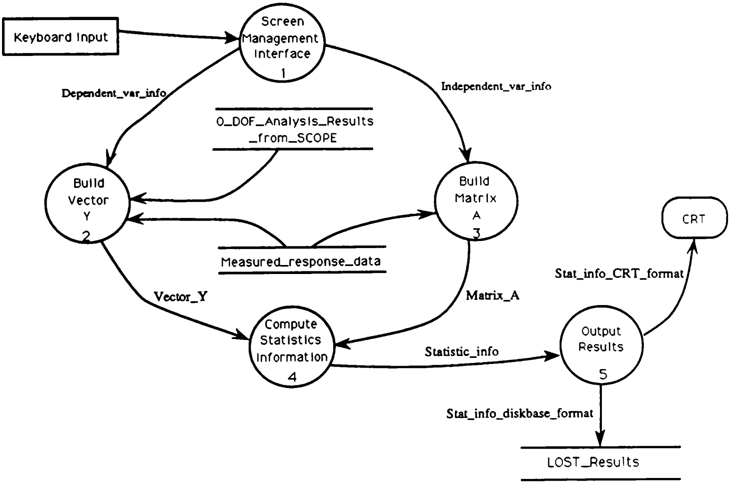
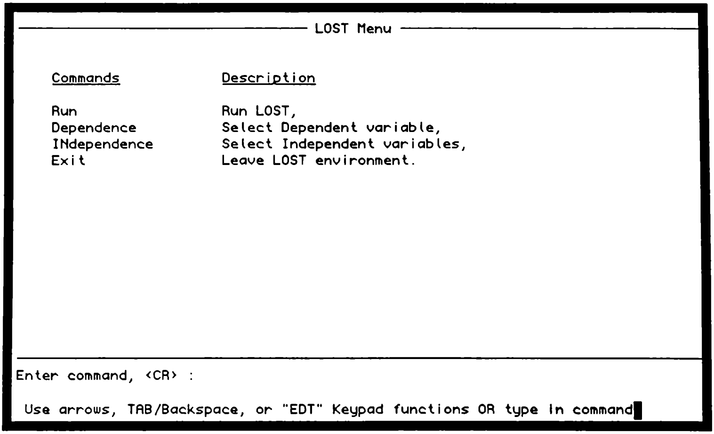
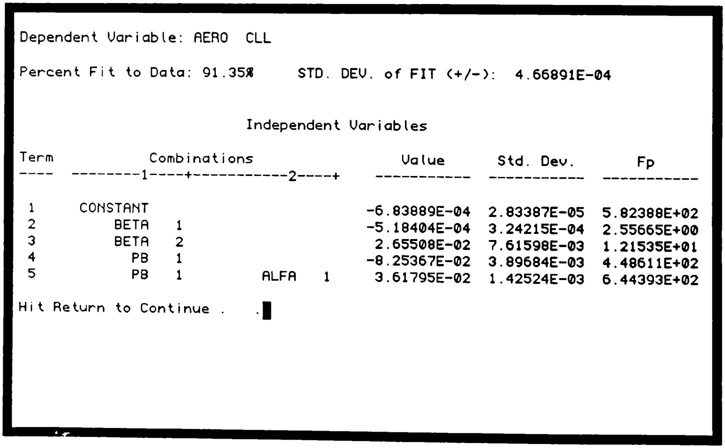


Figure 1. LOST high-level data flow diagram



```
---------------------------- LOST Menu ----------------------------

    Commands             Description

    Run                  Run LOST,
    Dependence           Select Dependent variable,
    INdependence         Select Independent variables,
    Exit                 Leave LOST environment.




---------------------------------------------------------------------
Enter command, <CR> :

 Use arrows, TAB/Backspace, or "EDT" Keypad functions OR type in command█
```

Figure 2. Main LOST menu



```
Dependent Variable: AERO  CLL

Percent Fit to Data: 91.35%     STD. DEV. of FIT (+/-):  4.66891E-04


                          Independent Variables

Term           Combinations                   Value       Std. Dev.        Fp
----   --------1----+-----------2----+    -----------  -----------  -----------

 1     CONSTANT                          -6.83889E-04  2.83387E-05  5.82388E+02
 2         BETA  1                       -5.18404E-04  3.24215E-04  2.55665E+00
 3         BETA  2                        2.65508E-02  7.61598E-03  1.21535E+01
 4           PB  1                       -8.25367E-02  3.89684E-03  4.48611E+02
 5           PB  1         ALFA    1      3.61795E-02  1.42524E-03  6.44393E+02

Hit Return to Continue .    .█
```

Figure 3. Example of LOST results as displayed on screen

RESULTS

Two examples of the LOST tool are demonstrated in this section. In Example 1, LOST is used to check the magnitudes of aerodynamic coefficients in a high-performance aircraft simulation. In Example 2, parameter increments determined using LOST are shown to improve the performance of an aircraft simulation.

Example 1 - Database Validation Using LOST

Incremental parameters representing errors in modeled aerodynamic derivatives (C_{l_β}, C_{l_P}, etc.) were estimated from actual flight test data using the LOST tool. This was accomplished by using LOST to estimate the errors a simulation model had in predicting actual aircraft force and moment coefficients. In this type of an analysis, the dependent variables are the time histories of the differences between measured and computed force/moment coefficients. These errors are postulated to be functions of the aircraft states and controls of the form:

$$\Delta Ca = \Delta C_{a_o} + \Delta C_{a\beta}\,\beta + \Delta C_{ap}\,P + \Delta C_{aR}\,R + \Delta C_{a\delta A}\,\delta A + \Delta C_{a\delta R}\,\delta R \;; \qquad a = Y, l, n$$

The independent variables in this analysis (β, P, R, δA, δR) are known from flight test data measurements. The incremental parameters ($\Delta C_{a\beta}$, ΔC_{ap}, etc.) were estimated using the LOST software.

Illustrated in Table 1 are the comparisons of the stability and control parameters in the simulation, the simulation parameters as augmented by the LOST increments, the stability and control parameters defined from flight test data and wind tunnel analysis of this aircraft. The flight test results were obtained using modern parameter estimation techniques applied to a wide variety of test data. Both the flight test results and wind tunnel results are of very high quality (as demonstrated by their ability to accurately predict actual aircraft response as compared to the simulation model).

LOST indicated that many of the parameters that were estimated were not statistically significant, therefore, they did not need to be changed. This find agrees well with the flight test and wind tunnel results, which also indicate that these terms are in generally good agreement. As can be seen by this analysis, the LOST-augmented estimates agree very well with the flight and wind tunnel results. This result indicates that LOST can effectively isolate the problems with the simulation model.

Example 2 - Use of LOST Estimates in Improving Simulation Fidelity

It was demonstrated in Example 1 that the LOST tool could effectively indicate the magnitude of the corrections needed to improve a simulation model. It would be of great benefit to a simulation engineer if the LOST estimates could be used to improve simulation fidelity. An example of this use is presented here. LOST estimates for the rudder force term increment ($\Delta C_{Y_{\delta R}}$) are compared to estimates obtained from a rigorous, large scale parameter estimation effort in an effort to show that the LOST results are consistent with the results obtained from more advanced tools. This comparison is shown in Figure 4. As can be seen from this figure, there is good agreement between the LOST results and the parameter estimation results.

In an effort to improve the simulation fidelity, the LOST results were used to augment the simulation model. Comparisons of the predicted C_Y coefficient are made between the old model, the model with the LOST correction, and the actual flight test lateral force coefficient are shown in Figure 5.

Table 1 Comparison of various parameters with LOST results

term	existing simulation value	LOST augmented simulation value	flight test value	wind tunnel value
C_{Y_β}	-.018	-.007	-.010	**
$C_{Y_{\delta R}}$	.0043	.0026	.0029	**
C_{l_β}	-.003	-.002	-.0022	-.0019
C_{l_P}	-.40	*	-.42	-.37
C_{l_R}	.125	*	.11	.11
$C_{l_{\delta A}}$	.0022	*	.0025	**
$C_{l_{\delta R}}$	.00030	*	.00027	**
C_{n_β}	.0018	.0011	.0011	.0015
C_{n_P}	-.02	-.14	-.05	-.05
C_{n_R}	-.17	*	**	-.15
$C_{n_{\delta R}}$	-.0014	*	-.0015	**

* - no change, LOST predicts change is statistically insignificant

** - not available

It can be seen from Figure 5 that the use of the LOST estimates greatly improves the force coefficient predictions of the simulation model.

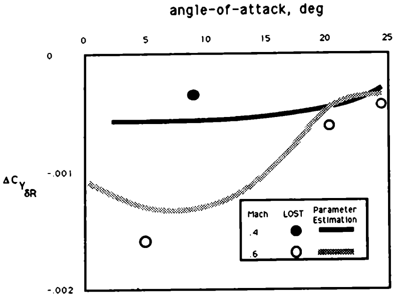


Figure 4. Comparison of increments to $C_{Y_{\delta R}}$

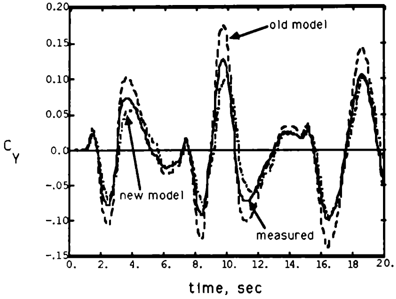


Figure 5. Comparison of measured and predicted lateral force coefficients

CONCLUSIONS

Modern techniques for simulation validation do not address the problem of isolating detailed problems with simulation models. The use of a simple estimation scheme (based on least-squares estimation) has been installed in a simulation environment. Though simple in nature, and easy to implement, such a scheme is very successful in isolating deficiencies in simulation models as well as identifying corrections to simulation models.

REFERENCES

1. Hess, R. A., Stanka, B. H., Purdy, M., 'A Methodology for Simulation Validation using Optimal Time History Matching', AIAA 88-4617-CP, September 1988.

2. Hines, W. W., Montgomery, D. C., Probability and Statistics in Engineering and Management Science, John Wiley & Sons, 2nd ed., 1980.

FLIGHT RECONSTRUCTION TECHNIQUES FROM FLIGHT RECORDER DATA FOR SIMULATION AND TRAINING

K. Krishnakumar[1], J. E. Bailey[2], and R. K. Prasanth[3]
The University of Alabama, Tuscaloosa, Alabama.

ABSTRACT

The concept of using limited "black-box" flight data for simulator flight reconstruction is introduced and the requirements are discussed. An extended linearized continuous Kalman filter design is utilized as a state estimator flight simulator driver. Actual Kalman filter implementation utilizes a UH-60 helicopter non-linear simulation (linearized aerodynamics) for state propagation to facilitate easy transition between flight reconstruction and manual simulation. System performance in relation to flight recorder data limitations is discussed. The resulting reconstruction system is demonstrated using both simulated data and a UH-60 "black-box" derived data.

INTRODUCTION

Flight simulation is used extensively for aircraft training and research in both civilian and military applications. One area where flight simulation has not been used extensively is flight reconstruction from measured flight data (data from the on-board instrumentation or from other means such as GPS translator and RADAR tracking). Flight reconstruction for simulator playback has several benefits in training and research, such as the following: (1) Flight simulation recreates the visual/motion/instrument displays of the recorded flight, permitting re-examination of a training flight in detail. (2) Flight recreation/simulation helps in understanding pilot behavior and error patterns and can be used for training, debriefing, and tactics analysis. (3) Recreation of flight for several aircraft simultaneously involved in an exercise is feasible and beneficial for combat training. (4) Flight mathematical model analysis and identification, handling qualities verification, pilot performance and scoring, and accident investigation are other areas where flight reconstruction/simulation can be used to advantage.

The increasing use of solid state flight data recorders in modern aircraft offers the engineer and the scientist an opportunity to acquire flight data on a routine, operational basis. However, most flight data recorders are still limited in instrumentation channels and data quality when compared with flight test instrumentation. Modern state estimation algorithms can be utilized to regenerate unmeasured states, thereby overcoming the problem of limited instrumentation channels. In this paper, optimal estimators (instead of optimal smoothers) are utilized for data playback due to the requirement for real-time playback and simulator-based algorithm implementation considerations. Moreover, state estimation methods are attractive because the measurement errors are optimally filtered by algorithms knowledgeable of the vehicle dynamics. Implementation of state estimation algorithms for flight simulator reconstruction and playback of a flight poses the following challenges and difficulties: (1) Most of the present flight data gathering devices are limited both in achieving fast sample rates and in obtaining error-free data. Due to this reason the visual flight reconstruction requires relatively sophisticated techniques for generation of the playback graphics which are both pleasing to the viewers and accurate in depicting the cockpit window scenes. (2) Recreation of motion and cockpit instrument values requires accurate estimation of the acceleration vector and other aircraft variables. Atmospheric disturbances such as wind and turbulence velocities may also be of interest. (3) Realistic "aircraft-like" maneuvers during playback require that measurement-smoothing be correlated well with the kinematic and dynamic motion of the aircraft. Therefore an accurate aircraft mathematical model is essential to the playback and reconstruction process.

This paper reports the results of a University of Alabama Flight Dynamics Laboratory analysis and implementation of a prototype, real-time, flight simulator-based reconstruction-playback system. A description of flight data recorders and their capabilities and the requirements for flight simulation/reconstruction in terms of both visual simulation

[1] Visiting Assistant Professor, Department of Aerospace Engineering, Member AIAA.

[2] Professor, Department of Aerospace Engineering, Member AIAA.

[3] Graduate Student, Department of Aerospace Engineering, Student Member AIAA.

and a simulator replay implementation are discussed. A new concept of tailoring a continuous extended Kalman filter estimator algorithm to an available flight simulation system for flight reconstruction is presented. Problems associated with observability and stability of the Kalman filter estimator and their relation to (1) choice of measured variables, (2) non-linear, time-variant Kalman filter implementation, and (3) robustness are discussed. A UH-60 helicopter mathematical model with both simulated flight recorder data and a limited "black-box" flight recorder-derived data is utilized for system demonstration.

REQUIREMENTS OF FLIGHT RECONSTRUCTION FOR FLIGHT SIMULATION (FRFFS)

The primary requirement for the FRFFS system is the "use of flight recorder data for driving a flight simulator to provide full flight reconstruction and replay for training application." Figure 1 depicts general components of a training simulator facility. In a typical training simulator, the vehicle mathematical model implemented is used to drive the motion, visual, and instrument systems. This study assumes generic flight simulation hardware, software, and mathematical model implementations. This assumption enables the design of the FRFFS to be easily adapted to other simulation facilities.

The training objectives of the FRFFS are as follows.

1. The FRFFS used for pilot training research should provide a student or his instructor the ability to study a recorded training flight in two modes of recreation: (a) visual-only simulation recreation of the recorded data or (b) total simulation recreation, including motion, visual, and instrument system simulations.
2. The FRFFS, in addition to recreating the recorded flight, should provide the student or instructor full manual control of the aircraft in a demonstration refly mode of operation.

The requirements of FRFFS system design can now be stated to meet the training objectives with emphasis on visual, motion, and instrument systems. Requirements are developed for two distinct categories:

1. Use of recorded aircraft variables to accurately recreate only the visual scenes.
2. Use of recorded variables to accurately recreate a complete flight simulation including a capability to manually control the flight.

Visual Scene Recreation Requirements

Replay of pilot or external observer visual scenes for a recorded flight requires, as a minimum, the following visual-scene related variables:

1. Rotational positions (Euler angles Θ, Φ, Ψ).
2. Translational positions (Inertial positions XI, YI, ZI).

If the flight data recorder measurements include XI, YI, ZI, $\Theta, \Phi,$ and Ψ, it is apparent that to recreate only the visual scenes the simulation aircraft model is not necessary. For this most simplistic visual recreation of the flight, the following requirements can be stated.

1. The flight data recorder measurements should include XI, YI, ZI, $\Theta, \Phi,$ and Ψ.
2. The kinematic relationship between XI, YI, ZI, $\Theta, \Phi,$ and Ψ should be maintained.
3. A smoothing algorithm, which filters the noise in the measurements and depicts a visual scene which is pleasing to the eye, has to be employed for good visual fidelity.
4. The visual scenes should be updated at a minimum rate of 15 Hz (> 30 Hz desirable).

When the flight data recorder measurements do not include any of the six kinematic states, the reconstruction process has to include an aircraft model.

Total Simulation Recreation Requirements

Referring to Figure 1, requirements for driving the simulator systems with flight recorder data can be stated as follows:

1. The non-linear simulation model used for the flight simulator is essential in maintaining the intended fidelity of the simulator. The FRFFS process should make use of this model for reconstruction to enable smooth and easy transition between FRFFS and manual flight simulation.
2. Estimates of all aircraft variables needed to drive the simulator cab systems, namely, the motion, visual, and instrument systems, must be generated.

The main focus in this paper is on the total simulator flight recreation problem. The visual-only flight recreation process is considered as a subset of this problem.

FLIGHT DATA PREPROCESSING

Digital flight data recorders (FDR) are capable of recording several measurements available from on-board and external sources such as the Global Positioning

Satellite systems (GPS) and radar tracking. Although digital flight data recorders provide data with less noise than their analog counterparts, the data set is generally not complete enough to fully describe the motion of an aircraft. In addition to the incompleteness of the data, the available data might have the following deficiencies: (a) Data dropout: Data coming in a stream might be disrupted for some reason, causing a void in the data stream for some period of time. (b) Data resolution: Depending on the digital recording format, the data may have poor resolution due to round-off errors. (c) Data sample rate: All data are recorded at predefined sample rates. Low sample rates lead to aliasing errors. (d) Data skewness: Data recorded at the same sample rates might be recorded at different time slots. Thus, the data must be synchronized before being used.

As an example, the measurement set used for UH-60 recorders in the FRFFS demonstration is presented in Table 1. The "frequency" defined in the table reflects the sample rate for each measurement. The "word slots" define the frame time slot in which the measurement is actually recorded. Each second is divided into 64 such slots. The "time-delay" parameter defines the time delays between a measurement and a reference measurement (chosen as UTM-northing in this study). These time delays can then be used for synchronization and data interpolation.

The FDR derived data was preprocessed to obtain a data stream to drive the real-time Kalman filter estimator. If the measurements include inertial positions and Euler angles, then this preprocessing can be used to generate the visual-only flight recreation mode. The preprocessing implemented in this study performs the following operations.

1. Data cleaning: eliminates unwanted and unusable data.
2. Data deskewing: synchronization of the data.
3. Data interpolation: low sample rate data is interpolated to obtain higher sample rate data.
4. Data smoothing: due to resolution limitations in some of the measurements, a smoothing algorithm is utilized.

The data interpolation is conducted using a cubic spline interpolation technique. Reference 1 presents a complete description of the algorithm needed to implement a cubic spline interpolation. The data smoothing is achieved by limiting the first and second order derivatives in the cubic spline interpolation.

FLIGHT RECONSTRUCTION USING KALMAN FILTER ESTIMATION

Several studies have been conducted in the areas of post-flight estimation and flight reconstruction using some form of a Kalman Filter (KF) [see Ref. 2 to 6]. Although these studies address some of the issues discussed herein, none discusses the FRFFS problem. The FRFFS problem is approached in this study as a classical estimation problem using a KF implementation, and two steps are taken to achieve the end result of simulator replay of a recorded flight. In step 1, the FDR data is preprocessed, as discussed earlier, and then sampled at the 30 Hz simulator frame rate. In step 2, a non-linear Kalman filter estimation is designed and built into the already existing simulation. Before discussing the design of the Kalman-filter estimation problem, a general form of a simulation mathematical model is presented.

General Form of a Simulation Math Model

A generalized body-axis mathematical model commonly implemented in flight simulators is presented below.

$$\dot{X} = f(X,U) + \text{Process Noise}$$

where f(X,U) is a function with non-linear terms due to aerodynamics, control system dynamics, and kinematics. The system state X is defined as

$$X^T = [\ XB^T \quad XC^T \quad XK^T \quad XG^T]$$

XB^T = [u v w p q r] = Body axis velocities.

XC^T = Control system states.

XK^T = [$\Theta\ \Phi\ \Psi$ XI YI ZI] = Euler angles and inertial positions.

XG^T = [ug vg wg pg qg rg] = Gust and wind states.

The control vector U is usually the pilot stick inputs. The non-linear kinematic equations are given by

$$\begin{bmatrix} \dot{\Theta} \\ \dot{\Phi} \\ \dot{\Psi} \end{bmatrix} = \begin{bmatrix} 0 & \cos\Phi & -\sin\Phi \\ 1 & \sin\Phi\tan\Theta & \cos\Phi\tan\Theta \\ 0 & \sin\Phi\sec\Theta & \cos\Phi\sec\Theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}$$

$$\begin{bmatrix} \dot{XI} \\ \dot{YI} \\ \dot{ZI} \end{bmatrix} = \begin{bmatrix} \cos\Theta\cos\Psi & \sin\Theta\sin\Phi\cos\Psi - \cos\Phi\sin\Psi & \cos\Phi\sin\Theta\cos\Psi + \sin\Phi\sin\Psi \\ \cos\Theta\sin\Psi & \sin\Theta\sin\Phi\sin\Psi + \cos\Phi\cos\Psi & \cos\Phi\sin\Theta\sin\Psi - \sin\Phi\cos\Psi \\ -\sin\Theta & \sin\Phi\cos\Theta & \cos\Phi\cos\Theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

Kalman Filter Estimation

The design and performance of an estimator for FRFFS based on the KF theory depends on (1) the system model, (2) the measurements and their noise characteristics, and (3) the observability and disturbability of the system. The system model used in flight simulators is usually non-linear. The non-linearities mainly arise due to (1) variations in aerodynamic coefficients due to varying airspeed and (2) non-linear kinematics due to heading angle changes (assuming the pitch and roll angles are within the linearity assumption; see Equation above). If either the extended or the linearized Kalman filter is used, the resulting estimation will be complex, and both time and computer-memory expensive. Therefore this approach is not very attractive for the real time FRFFS. A novel way by which a steady-state Kalman filter (SKF) designed for a linearized system could be used with the non-linear simulation for the FRFFS implementation is presented below.

Problem Statement:

Given a non-linear system, as defined earlier, and a measurement process Z as:

$\dot{X} = f(X,U) +$ Process Noise

$Z = h(X) +$ Measurement Noise

design a SKF gain matrix "KK" such that the resulting estimator will be

$$\dot{\hat{X}} = f(\hat{X},U) + [KK]\ (Z-h(\hat{X}))$$

Solution:

Part 1: The non-linear dependency on airspeed is eliminated by linearizing the system at discrete trim level flight airspeed points (Va_i, i=1..N), about a trim level flight. The resulting system model is

$\dot{x} = [F_i]\ x + [G_i]\ u +$ Process Noise; i= Airspeed index

$Z = [H_i]\ x +$ Measurement Noise

Where $x = X-X_i$; $u = U-U_i$; $[F_i]$, $[G_i]$, and $[H_i]$ are linear constant matrices; X_i and U_i are state and control trim vectors respectively.

The SKF gain matrices $[KK_i]$, for i = 1,.., N, are designed assuming the linear time invariant system model presented above (see Appendix A for a general definition of the SKF problem). Based on the above gains, the resulting flight variable estimator will be

$$\dot{\hat{X}} = f(\hat{X},U) + [KK]\ (Z-h(\hat{X}))$$

where the SKF gain matrix [KK] is a linearly interpolated matrix, based on the estimated airspeed, given as

$$[KK] = [KK_i] + \frac{[KK_{i+1}] - [KK_i]}{Va_{i+1} - Va_i}\ (\hat{Va} - Va_i)$$

Part 2: The above linearization assumption is valid for maneuvers where the small angle approximation is valid. During a normal flight, however, the heading angle goes through large changes. Large heading angle changes coupled through the non-linear kinematic equations induce effective system gain changes which make a linearized Kalman filter design invalid. Thus a Kalman filter designed for one particular heading angle becomes invalid for other heading angles (in some cases the resulting Kalman filter system will become unstable). In Appendix B, an analytical transformation is derived by which the Kalman filter gains designed for a reference heading angle (Ex: $\Psi = 0.0$) can be used for any other heading angle through a simple heading angle transformation. Referring to Appendix B, the final form of the estimator can be written as:

$$\dot{\hat{X}} = f(\hat{X},U) + [\ \Psi 1][KK][\ \Psi 2]^{-1}\ (Z - h(\hat{X}))$$

Figure 2 shows a conceptual block diagram of the SKF flight estimation implementation. The main advantages of the above scheme are off-line filter gain computation and the substantial reduction in computational requirements. The number of computations per iteration required for an extended Kalman filter gain computation is proportional to the third power of the filter dimension. In the FRFFS design presented above, this computation is carried out off-line.

Observability, Stability, and Robustness

The necessary and sufficient conditions for the Kalman filter stability, assuming perfect knowledge of a priori conditions, are complete observability (through the measurements) and complete disturbability (through the system disturbances) [Ref. 7]. Complete disturbability can be achieved by adding pseudo noise to the undisturbed modes. On the other hand, complete observability depends entirely on the measurements available. Table 2 presents combinations of measurements which provide observable and unobservable systems. The last column presents the observability indices for the respective measurement set. These indices are interpreted as the number of integrations needed to completely observe the state vector. An observability index of zero means that all the states are measurements. It is interesting to note that the addition of airspeed as one of the measurements greatly decreases the observability index. The bandwidth of the estimator system increases with decreasing observability index.

The filter performance might also exhibit "apparent divergence" (estimate errors are large but bounded) if there are model errors, unestimated biases, and errors in control input measurements. One way to avoid an apparent divergence is by designing a robust estimation system. The robustness of the estimator depends greatly on the measurement uncertainties. Reducing the measurement noise covariance results in a system with large SKF gains. Large filter gains increase the stability of the system. Robustness can also be achieved by designing the SKF gains using modal destabilization techniques [Ref. 4]. For the FRFFS problem the importance of having a robust system is significant due to the certainty of errors in the assumed models of the aircraft, measurements, and control system.

FRFFS EXAMPLES

A UH-60 non-linear helicopter mathematical model is utilized (Ref. 7) to demonstrate the FRFFS concept. Two examples, one based on data obtained from a simulated UH-60 flight and the other based on data obtained from a FDR, are presented. The UH-60 simulation utilizes linearized aerodynamics and non-linear kinematics. Table 3 presents the description of the 23 states used in the UH-60 simulation and the SKF design. It is assumed that the control inputs are perfect measurements and are used without filtering. Two measurement schemes, one with airspeed and the other without, are used to demonstrate the effect of airspeed as an additional measurement. Table 4 presents the measurement set and the noise characteristics of the respective measurements used in the SKF design. The choice of these measurements is based on the UH-60 FDR measurement set given in Table 1.

Example 1: For this example, a measurement data set was recorded, as per specifications given in Table 1, from a simulated UH-60 flight. This example illustrates a situation in which non-linearity in heading angle changes and airspeed variations is present, but there are no modeling errors. The FRFFS implementation presented in Figure 2 is utilized for flight replay. Figures 3 through 5a present data for attitudes (Φ, Ψ), inertial positions (XI-YI, ZI), and airspeed (Va).

The good tracking characteristics of the estimator are apparent from Figures 3 through 5a. The major difference in performance between the two estimator systems is in gust state estimation. Figure 5b presents longitudinal gust estimates for a simulated flight flown in turbulence. This figure indicates that when airspeed is included as a measurement the estimator system bandwidth is increased considerably. While the estimator based on six measurements is unable to estimate gust states due to its low bandwidth, the filter that uses airspeed estimates gust satisfactorily.

Example 2: In this example, a measurement set obtained from a UH-60 flight data recorder (FDR) is utilized for FRFFS demonstration. As in the first example, the FRFFS implementation shown in Figure 2 is utilized. Airspeed was not considered as a measurement due to the poor quality and large data dropouts in the FDR data. The FDR data also showed poor resolution in the inertial position measurements, with the round-off error in XI and YI being of the order of 30 feet. Figure 6 presents estimated data for XI-YI, airspeed, and heading angle. This figure shows that the SKF design is able to estimate the aircraft states without exhibiting any apparent divergence, in spite of the following error sources in the suboptimal filter design: 1) aircraft modeling errors, 2) control input biases, 3) non-linear effects due to large roll and pitch angle changes, and 4) errors in the assumed error covariances for the measurements.

CONCLUDING REMARKS

A methodology by which a limited FDR derived data set can be utilized for reconstructing a flight for simulator recreation was presented. Steady-state suboptimal Kalman filter design with linear interpolation for varying aerodynamics and heading angle transformation is utilized in implementing the non-linear estimation algorithm. Examples of flight reconstruction using simulated data and "black box" data were illustrated. It was shown that the inclusion of airspeed as a measurement improves the bandwidth of the resultant system. The FRFFS implementation displayed robust characteristics in tracking the limited UH-60 flight data recorder data.

Several areas remain in which further research is needed to improve the performance and robustness of the FRFFS design, such as a study to determine the optimal measurement set and measurement rates required for the FRFFS implementation. This information could then be used to design future flight data recorders. Robustness with respect to model errors, control input estimation, self-adaptive estimation, and model reduction techniques are other areas relevant to the FRFFS problem.

REFERENCES

1. Rice, J. R., Numerical Methods, Software, and Analysis, McGraw-Hill Book Company, New York, 1983.

2. Bach, R. E. and Wingrove, R. C., "The Analysis of Airline Flight Records for Winds and Performance with Application to the Delta 191 Accident," AIAA Paper 86-2227, Williamsburgh, Va, 1986.

3. Bach, R. E. and Wingrove, R. C., "Applications of State Estimation in Aircraft Flight-Data Analysis," Journal Of Aircraft, July 1985, pp 547-554.

4. Bryson, A. E., Jr., "Kalman Filter Divergence and Aircraft Motion Estimators," Journal of Guidance and Control, Jan-Feb 1978, pp 71-79.

5. Verhaegen, M. N., "Robust Adaptive Flight-Path Reconstruction Technique for Nonsteady Longitudinal Flight Test Maneuvers," Journal of Guidance, Control, and Dynamics, Jan-Feb 1988, pp 72-79.

6. Wingrove, R. C., "Quasilinearization Technique for Estimating States from Flight Data," Journal of Aircraft, Vol. 10, May 1973, pp 303-307.

7. Maybeck, P. S., Stochastic Models, Estimation, and Control, Vol. 12, Academic Press, New York, 1976.

8. Hilbert, K. B., "A Mathematical Model of the UH-60 Helicopter," NASA TM-85890, April, 1984.

9. Gelb, A., Applied Optimal Estimation, MIT Press, Cambridge, Mass., 1974.

10. Whorton, M. S., "Mathematical Modeling and State Estimation for a UH-60 Flight Data Replay and Refly System," M. S. Thesis, The University of Alabama, March, 1989.

11. Bailey, J. E., Krishnakumar, K., and Prasanth, R. K., "UH-60 Simulator Playback Kalman Filter Design and Implementation," The University of Alabama BER Report No. 455-177, March, 1989.

APPENDIX A: CONTINUOUS STEADY STATE KALMAN FILTER EQUATIONS (Ref. 9)

System model:

$$\dot{X}(t) = F\,X(t) + L\,U(t) + G\,W(t)\ ;\ W(t) \sim N(0,Q)$$

Measurement model:

$$Z(t) = H\,X(t) + V(t)\ ;\ V(t) \sim N(0,R)$$

Initial Conditions:

$$E[X(0)] = \hat{X}(0);\quad E[X(0)-\hat{X}(0)][X(0)-\hat{X}(0)]^T = P_0$$

Steady State Estimation:

$$\dot{\hat{X}}(t) = F\,\hat{X}(t) + L\,U(t) + [KK](\,Z(t)-H\,\hat{X}(t)\,)$$

Steady State Riccati Equation:

$$F\,P_\infty + P_\infty F^T + GQG^T - P_\infty H^T\,R^{-1}\,H\,P_\infty = 0$$

Steady State Kalman Gain Matrix:

$$[KK] = P_\infty H^T\,R^{-1};\quad \text{For}\quad E[W(t)V(t)\,] = 0.$$

APPENDIX B: KALMAN FILTER GAIN MATRIX HEADING ANGLE TRANSFORMATION

Let superscript * denote the quantities defined for a zero heading angle ($\Psi = 0$) and let variables with no superscript denote quantities defined for any heading angle (Ψ). Then the state vector (order=n), the measurement vector (order=m), and the system matrix (n X n) can be related as follows:

$$X = [\Psi 1]\,X^*\ ;\ Z = [\Psi 2]\,Z^*\ ;\ F = [\Psi 1]\,F^*\ ; \quad \text{--->}\ \text{B.1}$$

where

$$X^T = [\ u\ v\ w\ p\ q\ r\ .\ .\ .\ .\ .\ .\ .\ .\ XI\ YI\ ZI\ ]$$

$$Z^T = [\ .\ .\ .\ .\ XI\ YI\ ZI\ ]$$

$$[\Psi 1] = \begin{bmatrix} [\ I1\ ] & [\ \text{zero}\ ] \\ [\text{zero}] & [\ \ T_\Psi\ \ ] \end{bmatrix}$$

$$[\Psi 2] = \begin{bmatrix} [\ I2\ ] & [\ \text{zero}\ ] \\ [\text{zero}] & [\ \ T_\Psi\ \ ] \end{bmatrix} \quad \text{--->}\ \text{B.2}$$

$$[T_\Psi] = \begin{bmatrix} \cos\Psi & -\sin\Psi & 0 \\ \sin\Psi & \cos\Psi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

[I1] = (n-3) X (n-3) Identity matrix.

[I2] = (m-3) X (m-3) Identitiy matrix.

[zero] = Matrix of zeros with consistant dimension.

With the above definitions, the Steady state Kalman filter estimation equations for a zero heading angle and for a general heading angle are given by:

$$\dot{\hat{X}}^* = F^*\,\hat{X}^* + KK^*\,(Z^* - H\,\hat{X}^*) \quad \text{--->}\ \text{B.3}$$

$$\dot{\hat{X}} = F\hat{X} + KK(Z - H\hat{X}) \quad \text{--->} \ B.4$$

From the definitions presented in equation B.1, it can be shown that

$$F^{*}[\Psi1]^{-1} = F^{*} \quad \text{and} \quad H\,[\Psi1]^{-1} = [\Psi2]^{-1}H \quad \text{--->} \ B.5$$

Substituting the above equations in B.3 we have

$$\dot{\hat{X}} = F\hat{X} + [\Psi1]KK^{*}\,[\Psi2]^{-1}(Z - H\hat{X}) \quad \text{--->} \ B.6$$

which implies $KK = [\Psi1]KK^{*}\,[\Psi2]^{-1}$ ---> B.7

The above equation can be used to compute gains for varying heading angles without actually having to compute the solution of the time-varying Riccati equation.

Table 1 UH-60 Measurement Set Definition

Parameter	Frequency (Hz)	Word Slots (64/sec)	Time Delay* (In sec.)
UTM Northing (XI)			
Low Byte	4	2 18 34 50	0
High Byt	4	10 26 42 58	
UTM Easting (YI)			
Low Byte	4	5 21 37 53	0.04688
High Byte	4	13 21 37 53	
Roll Angle	8	8 16 24 32	
		40 48 56 64	0.09375
Pitch Angle	8	4 12 20 28	
		36 44 52 60	0.03125
Yaw Angle	8	6 14 22 30	
		38 46 54 62	0.06250
Altitude (-ZI)	8	3 11 19 27	
		35 43 51 59	0.01563
Airspeed	2	15 47	0.20313
Collective Stick	2	7 39	0.07813
Longitudinal Stick	2	31 63	0.45313
Lateral Stick	2	23 55	0.32813
Pedal	2	17 49	0.23438

* Referenced to UTM Northing low byte.

Table 2. Observability Through Measurements

Measurements	Observable	Observablity Index
Θ, Φ, Ψ, XI, YI, ZI	Yes	13
X, Y, Z	Yes	13
Θ, Φ, Ψ, XI, YI	No	..
XI, YI, ZI, Va	Yes	5
XI, YI, Va	No	..

Table 3. 23 rd Order Kalman Filter States

Category	States	Units	category	States	Units
	u	Ft/sec		X1	Inch
	v	Ft/sec	Stability	X2	Inch
Aircraft	w	Ft/sec	Augmentation	X3	Inch
Body-axis	p	Rad/sec	System (SAS)	X4	Inch
Velocities	q	Rad/sec	States	X5	Inch
	r	Rad/sec			
Euler	Θ	Rad.		ug	Ft/sec
Angles	Φ	Rad.		vg	Ft/sec
	Ψ	Rad.	Gust shaping	wg	Ft/sec
Inertial	XI	Ft.	Filter	pg	Ft/sec
Positions	YI	Ft.	States	qg	Ft/sec
	ZI	Ft.		rg	Ft/sec

Table 4. UH-60 Measurement Set

Measurement	Noise Covariance σ^2	Unit
Airspeed (Va)	25.00	Ft^2/sec^2
Pitch Angle (Θ)	0.0028	Rad^2
Roll Angle (Φ)	0.0028	Rad^2
Heading Angle (Ψ)	0.0028	Rad^2
Inertial Positions (XI, YI, ZI)	100.0	Ft^2

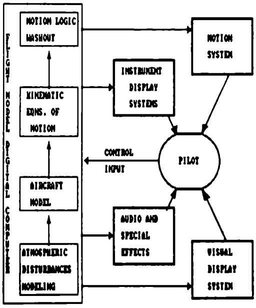

FLIGHT MODEL DIGITAL COMPUTER
MOTION LOGIC WASHOUT
KINEMATIC EQNS. OF MOTION
AIRCRAFT MODEL
ATMOSPHERIC DISTURBANCES MODELING
MOTION SYSTEM
INSTRUMENT DISPLAY SYSTEMS
CONTROL INPUT
PILOT
AUDIO AND SPECIAL EFFECTS
VISUAL DISPLAY SYSTEM

Fig. 1 Typical flight simulator components.

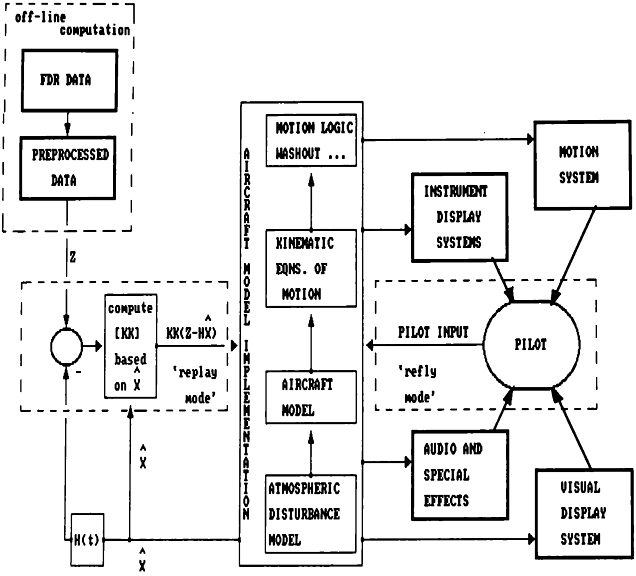


Fig. 2 Flight reconstruction ('replay')/ flight simulation ('refly') implementation block diagram.

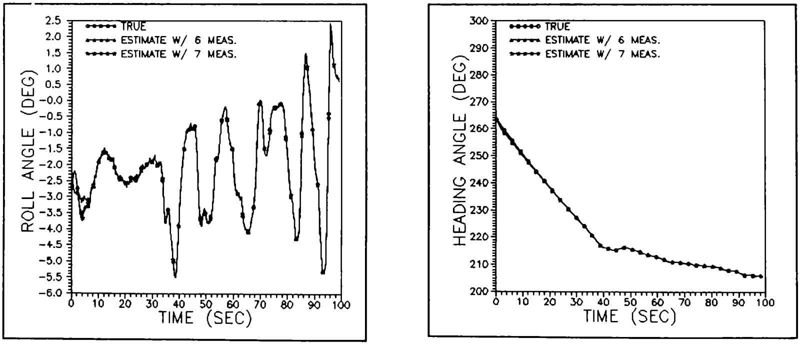


Fig. 3 True and estimated states for Example 1: a) roll angle and b) heading angle.

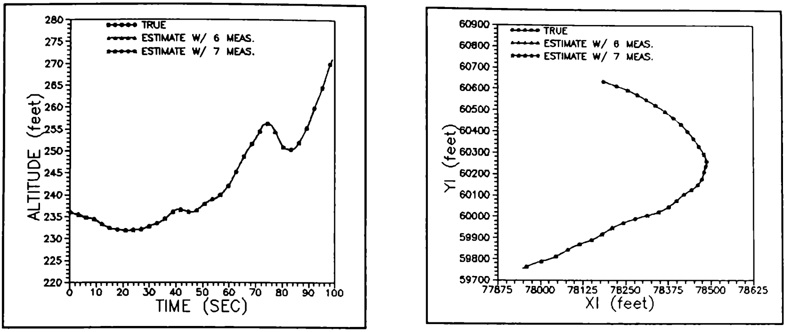


Fig. 4 True and estimated states for Example 1: a) altitude and b) ground track.

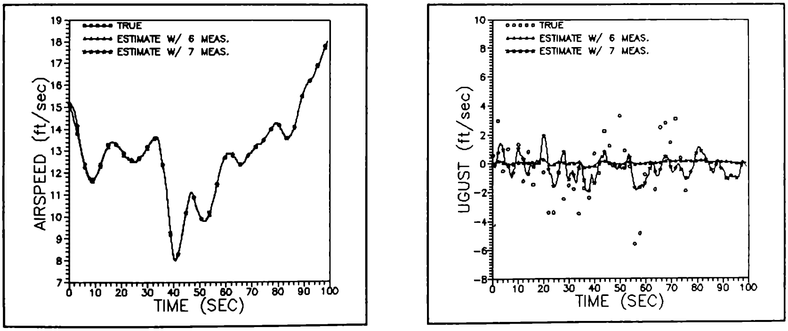


Fig. 5 True and estimated states for Example 1: a) airspeed and b) longitudinal gust velocity.

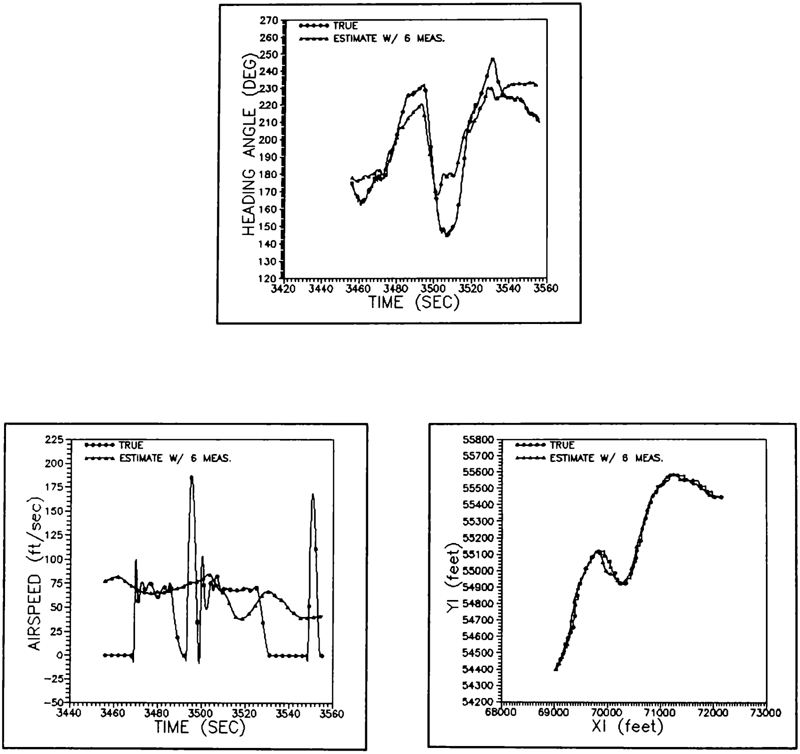


Fig. 6 FDR measurements and estimated states for Example 2: a) heading angle, b) airspeed, and c) ground track.

89-3264-CP

PAYLOADS SIMULATION IN THE SHUTTLE MISSION TRAINING FACILITY

Ankur R. Hajare and Patrick M. Brown*
MITRE
Houston, Texas

Abstract

The Shuttle Mission Training Facility (SMTF) at the NASA Johnson Space Center is the primary facility for full mission training of astronauts. The training provided in the SMTF includes payload operations that are performed by the astronauts while the Space Shuttle is in orbit. This training requires real-time simulation of the operational aspects of Shuttle payloads as well as simulation of the interaction between payloads and the Orbiter. For this purpose the SMTF contains Payloads Simulators (PLS) that are coupled to and operated in synchronization with the Shuttle Vehicle Simulators. The PLS are used for simulating all payloads with the exception of the European Space Agency's Spacelab. The SMTF contains a separate Spacelab Simulator for training Spacelab astronauts.

This paper provides a synopsis of actual payload operations, the methodology for payload simulation in the SMTF, and the system architecture of the PLS.

Introduction

The Shuttle Mission Training Facility (SMTF) provides astronaut training for Shuttle crew members which include a commander, a pilot, mission specialists, and payload specialists. Training covers all mission phases, including payload operations, and is accomplished in two Space Shuttle simulators, called the Fixed Base (FB) and Motion Base (MB) Shuttle Mission Simulators. A third simulator, called the Guidance and Navigation Simulator (GNS), is currently used for simulator software development and not for astronaut training. Each of these three is comprised of a Shuttle Vehicle Simulator (SVS) and a Payloads Simulator (PLS).

Payload training in the SMTF is complemented by payload training at other facilities, especially for payload specialists who are not career astronauts. However, the SMTF is the only facility that provides full mission rehearsals, training the entire crew working as a team in concert with ground support personnel.

Space Shuttle missions are monitored and controlled from the Mission Control Center (MCC) at the Johnson Space Center (JSC). Payloads are monitored and controlled from Payload Operations Control Centers (POCCs) at JSC and other NASA Centers. There is a POCC at JSC for attached payloads, another at the Jet Propulsion Laboratory for interplanetary payloads, and several POCCs at Goddard Space Flight Center for various satellites. The SMTF supports the training of personnel in these ground support facilities as shown in Figure 1. The SMTF contains a simulator, called the Network Simulation System (NSS), that replicates the operation of the Tracking and Data Relay Satellite System (TDRSS) and its ground station at White Sands, New Mexico. The NSS enables the payload operations personnel at the MCC to communicate with either the FB or the MB simulator, just as they would communicate with the Space Shuttle during an actual mission. This is referred to as an "integrated" simulation session. To train ground personnel for payload operations at a remote POCC, there are "joint-integrated" simulation sessions which include a remote POCC in addition to the MCC, the NSS and either the FB or the MB.

Space Shuttle Payloads

The Shuttle payload bay is 15 feet in diameter and 60 feet long. It was designed to carry up to 65000 pounds into an easterly low Earth orbit and to return 32000 pounds of cargo back from space. The payload bay is designed to hold securely a wide range of objects which may include communications satellites, an autonomous Spacelab for experiments in space, or scientific cargo mounted on special pallets.

* Member AIAA

Space Shuttle payloads belong to one of these two categories: (1) payloads that are deployed in orbit, and (2) payloads that remain attached to the Orbiter throughout the mission. The first category includes satellites for communications, weather, and surveillance; scientific payloads such as the Hubble Space Telescope; and space exploration payloads such as the Venus radar mapper, Galileo, and Ulysses spacecraft. Payload orbits range from low Earth orbit, through geo-synchronous, to interplanetary orbits. Attached payloads include scientific experiments whose instrumentation is operated by astronauts. These may be experiments in materials science, medicine or other life sciences. Also included in the category of attached payloads are the small "Getaway Specials" that are self-contained apparatus requiring little or no astronaut activity. The Spacelab, an attached laboratory built by the European Space Agency, is the only manned payload carried by the Shuttle.

Since the Shuttle only goes into low Earth orbit (150 to 200 miles above the Earth), booster rockets called upper stages are required for lifting payloads into higher orbits. The two main upper stages are the Payload Assist Module (PAM) and the Inertial Upper Stage (IUS). The PAM, also known as the Solid Spinning Upper Stage, is a single stage booster with a solid propellent and is spin stabilized. The IUS is a two-stage, solid-fueled, three-axis controlled, inertially navigated rocket. A more powerful, liquid-fueled upper stage, called Centaur, was adapted for use with the Shuttle. However, it was cancelled because of safety considerations following the Challenger accident.

The size of the payload bay permits the Space Shuttle to carry multiple payloads. For example, Mission 61B in November, 1985 had three satellites and two experiments designed to study space construction by astronauts during extravehicular activity.

Future payloads will become increasingly complex. Foremost among those planned are the components of the Space Station which will be assembled in orbit by astronauts. Another complex payload scheduled for 1993 is the Orbital Maneuvering Vehicle (OMV). It will function as a space tug either maneuvering other spacecraft from the Shuttle payload bay to a different orbit or retrieving expired satellites and docking with the Space Shuttle.

Interfaces between the Orbiter and the payload serve to attach the cargo to the Orbiter or provide services from the Orbiter to cargo items. These interfaces are mechanical, thermal, avionics, power and fluid systems.

Payload Operations

Payload operations performed on board the Shuttle include satellite deployment and the performance of various experiments. The Shuttle provides the unique capability of retrieval and re-deployment of spacecraft after on-orbit servicing or repair. The first example of this was the retrieval, repair and re-deployment of the malfunctioning Solar Maximum Mission spacecraft during Mission 41-C in 1984.

Payload operations are performed at the aft flight deck of the Space Shuttle where there are four work stations in a shallow "U", as shown in Figure 2. Astronauts can observe payload operations through the two payload bay windows and the two overhead windows. In addition, the Space Shuttle has a Closed Circuit Television (CCTV) system that is used for monitoring payload operations.

The Remote Manipulator System (RMS), i.e. the "arm", of the Shuttle is used in deploying, manipulating and retrieving payloads. The RMS[1], illustrated in Figure 3, consists of Shoulder, Elbow and Wrist joints connecting upper and lower arm booms, and a payload grappling device called the End Effector. The Shoulder, which is attached to the Space Shuttle, has two degrees of freedom, the Elbow has one and the Wrist has three, thereby giving a total of six control degrees of freedom.

The RMS is operated by an astronaut by means of a Rotational Hand Controller (RHC), a Translational Hand Controller (THC), and other controls located on a Display and Control panel. All these controls interface to the Manipulator Controller Interface Unit (MCIU) which is connected to the flight computers of the Space Shuttle.

The Shuttle has a CCTV camera at each end of the payload bay, a camera mounted on a tilt and pan unit on the Elbow of the RMS, and a camera with a light on the Wrist of the RMS[2]. At any one time an astronaut operating the RMS can view two CCTV images on monitors located beside the Display and Control panel.

The Orbital Maneuvering System (OMS) consists of rocket engines used for major maneuvers such as transferring to another orbit for rendezvous with other spacecraft. Certain payload operations, such as a PAM deployment, require the Orbiter to be in a specific orientation. This is performed by means of the Reaction Control System (RCS) which includes thrusters in the front and rear of the Orbiter. The RCS is also used for small maneuvers during proximity operations. For payload operations, these delicate maneuvers are performed by the pilot, using the controls at the aft flight deck.

Payload Simulation

Figure 4 shows the configuration of the FB simulator, the MB simulator and the GNS, each of which consists of a SVS connected to a PLS. The SVS consists of a Sperry 1100/92 host computer, a Concurrent Computer Corporation 3280 Base Intelligent Controller (IC), and simulation equipment. The PLS is connected to the Base IC via shared memory. Further description of the SMTF configuration can be found in [3].

Simulation Methodology

Payloads are simulated in the SMTF by mathematical models that run in real time. The simulation is restricted to the operational features of payloads since its sole purpose is to train astronauts. This methodology is in contrast to an engineering simulation where the purpose is to design or develop payloads and their operational procedures.

In the SMTF astronauts are trained for both normal and contingency operations including mission abort sequences. Such contingency procedures can include jettisoning a payload. Also, some payloads have sensors that can generate caution and warning signals which are transmitted via the Shuttle's caution and warning system. Consequently, the software includes a large number of simulated malfunctions that are activated by an instructor during simulation sessions. Thus, the crew learns not only the nominal payload deployment sequence but also rehearses a large number of malfunction responses.

The payloads simulation in the SMTF can be run only in conjunction with the Shuttle simulation, i.e., it is not capable of stand-alone operation. As can be seen in Figure 4, the PLS does not have its own Instructor/Operator Station (IOS). Instead, it is operated from the IOS of the SVS. Thus, the payloads simulation is constrained to be in the same mode (e.g. "run" or "freeze") as the SVS. The Spacelab Simulator, which is described in a later section, is an exception to this.

Payloads simulation in the SMTF is synchronized with the real-time simulation of the Space Shuttle which is executed in 40 ms frames, i.e. at 25 Hz. In order to maintain synchronization, payload models run at a frequency of 25 Hz or submultiples thereof. Synchronization is maintained by a timing pulse at the beginning of every frame.

There is close interaction between the Space Shuttle and its payloads and, therefore, between the SVS and payloads simulation. Payloads affect the vehicle dynamics of the Space Shuttle, e.g. deployment or berthing of satellites changes the mass properties of the Shuttle. This is simulated in real-time by computing the forces and moments induced on the Orbiter vehicle by the moving payload, as well as changes to the center of mass of the Orbiter.

Payloads are often supplied electrical power from the Orbiter. Therefore, power consumption by payloads is simulated in conjunction with the simulation of the electrical power system of the Space Shuttle. Certain payloads use the Orbiter's thermal management system to dissipate heat. Therefore, the heat dissipation for such a payload is simulated in conjunction with the thermal simulation of the Space Shuttle.

The RCS, which maintains the attitude of the Shuttle while in orbit, may operate automatically in response to payload operations. This is simulated, including the attendant sound effects and fuel consumption. Similarly, the use of the OMS for payload operations is also simulated.

The SVS has a duplicate of the Space Shuttle's Data Processing System (DPS) which consists of five IBM General Purpose Computers (GPCs). A custom-built Simulation Interface Device (SID) interfaces the GPCs to the Base IC. The actual flight software of the Space Shuttle is executed in real time during a simulation. Since payloads interact with both the hardware and software of the Space Shuttle[4], the flight software varies for each mission and is payload dependent.

Crew Stations

Of the three Space Shuttle simulators in the SMTF, the FB simulator is used primarily for payload training. The FB crew station is a faithful replica of the Shuttle flight deck, both forward and aft. The aft FB crew station is identical to the aft flight deck shown in Figure 2 and it contains all the controls and displays that are accessible to the astronauts, including the Display and Control panel of the RMS, the THC and the RHC. The keyboards and CRTs on the Shuttle flight deck are duplicated in the crew station and are coupled, through the simulator, to the GPCs executing flight software. The FB crew station is on an elevated platform. Entrance is through a hatch opening in the floor of the crew station, as in the actual vehicle.

The Motion Base crew station replicates only the forward part of the Shuttle flight deck. This base is used to train astronauts for launch, ascent, re-entry and landing, i.e. all mission phases besides the on-orbit phase. Therefore, it is not used to train astronauts for payload operations. An addition to the Motion Base simulator is the Motion Base Aft crew station which replicates the aft portion of the Shuttle flight deck. It provides additional training capacity for on-orbit operations.

The GNS is not presently equipped for crew training.

Visual Simulation

The visual scenes observed by the astronauts during payload operations are simulated using digital image generation. Out-the-window visual imagery is provided for the two payload bay windows and the two overhead windows. The visual scenes include payloads with the proper shape and size, the visible portions of the Space Shuttle and the RMS, and astronomical objects. Payload features pertinent to astronaut operations, such as grapple fixtures, are simulated with greater detail than the rest of the payload. Lighting from the sun, the payload bay lights, and the light on the RMS wrist is simulated along with the resultant shadows.

The CCTV system of the Space Shuttle is simulated with digital image generation produced by the same visual system as the out-the-window scenes described above. The simulation of the CCTV includes camera effects and controls, i.e. camera selection, tilt, pan, and zoom. Thus astronauts in the FB simulator can operate the CCTV controls and view images on the two CCTV monitors. The CCTV in the Shuttle and the simulator are both monochrome and have the same video format.

RMS Simulation

The RMS, with all its controls, is simulated by software models. The controller between the "arm" and the GPCs, called the Manipulator Controller Interface Unit (MCIU), is functionally modelled. Simulating the MCIU includes replication of its interaction with the DPS of the Orbiter. The astronauts are provided visual feedback of RMS operations via out-the-window scenes and CCTV images.

Figure 3 illustrates RMS simulation in the SMTF. Astronauts in the simulator crew station operate the RHC, THC and other controls. These inputs are fed to the Signal Conversion Equipment (SCE) which performs analog-to-digital conversions for inputs and digital-to-analog conversions for outputs. RMS models resident in the SVS host computer interact with payloads models in the PLS. Both provide synchronized inputs to the visual system that generates the displays for the overhead windows, the payload bay windows and the CCTV.

The RMS is designed to handle payloads as large as 65000 lbs., i.e. the payload capacity of the Shuttle. Massive payloads can cause considerable flexing of the arm but the RMS simulation in the SMTF is kinematic. Bending of the upper and lower booms and the resultant oscillations are currently not modelled due to limited processing power. However, mission specialists receive additional RMS training at the Remote Manipulator System Simulation Facility (SIMFAC) in Toronto which provides a dynamic simulation that includes bending of the upper and lower booms[5].

Simulation of Payload On-Board Computers

Payloads often have their own on-board computers which are space rated minicomputers or microprocessors. For example, Spacelab has a set of on-board computers built in France by Matra. Different simulation methods have been used in the SMTF for replicating the functions of payload computers. Usually they are functionally simulated, as in the case of the on-board computers of Spacelab. However, for the simulation of the five Z-80 microprocessors that controlled the Centaur upper stage booster, commercial Z-80 microprocessors were procured. In this case, flight hardware was duplicated because it was cheap and easily available. This is often not the case, especially with custom-built processors used by DOD. A third method that has been used is instruction set emulation of the flight computers. This method is preferable to functional simulation when the source code is not easily available or when the payload's flight software is undergoing frequent changes.

Payload Telemetry

During "integrated" simulation with the SMTF, the MCC sends and receives a data stream just like it would during a real mission. Similarly, during "joint-integrated" simulations, the POCCs send and receive a data stream just as they would during a real mission. For certain payloads, a simulated telemetry stream from the SMTF is sent to the Satellite Control Facility (SCF) in California.

The NSS in the SMTF generates a data stream that contains simulator data (from either the FB or the MB simulator) in the same format and protocol as the real-world data streams. Similarly, the NSS receives and forwards the simulated uplink data streams from the MCC and a POCC. In addition to data from the mathematical models of the Shuttle systems, the data stream contains simulated CCTV imagery as well as digitized voice from the personnel participating in the simulation.

Both Shuttle and payload data are interleaved in the downlinked data stream. The NSS receives data from both the PLS and the SVS and integrates it into one simulated downlinked data stream. The uplinked data stream contains payload commands. The NSS extracts these and provides them to the PLS which performs the appropriate operations on the payload models.

Payload telemetry is relayed through the Orbiter until deployment of the payload and, in some cases, for a short period after deployment. The Orbiter does not have continuous communications with the ground throughout a mission. At certain points in its orbit there is a "Loss of Signal" (LOS) and communications with the ground is lost until an "Acquisition of Signal" (AOS) occurs. This was a frequent occurrence before there were two TDRSS satellites in operation. The Orbiter has a recorder specifically for recording payload data when it cannot be transmitted to the ground[6]. The NSS simulates LOS and AOS. It records and plays back the simulated payload data stream as it would be done after an LOS and AOS during a real mission. Because payload models in the SMTF are operations oriented, the payload data stream often does not contain meaningful data but instead, dummy data is inserted into the simulated data stream using the proper protocol.

Generalized Payload Model

Many simple payloads require astronauts to operate a set of switches in response to certain conditions and in a specific sequence. Such requirements are often stated in the form of logic diagrams. A software process, called Generalized Payload Model (GPM) was built to easily translate logic diagrams of payload operational sequences into simulator software.

Reuse of Payloads Software

Space Shuttle payloads are sometimes repeated with relatively small changes. For example, TDRS-A was put into orbit in mission STS-6 and TDRS-B was lost in the Challenger accident. TDRS-C was put into orbit by STS-26 and TDRS-D by STS-29. Such a repetition of payloads provides an excellent opportunity for the reuse of payload simulation software. This is particularly true of upper stages. The IUS has been used multiple times in the past and the flight manifest calls for its use six more times for unclassified missions by 1990. Therefore, IUS software written for STS-6 has been and will continue to be reused with appropriate modifications.

Since the payloads change on each Shuttle mission, new simulation software loads must be built to prepare for each mission. These loads contain simulation software for each payload for which training is required in the SMTF. In the case of payloads that are repeated, old simulation software is reused with appropriate modifications. When loads are generated, this reused software is integrated with software written for the new payloads.

The Simulator Reconfiguration System (SRS), shown in Figure 4, was developed to support an increasing flight rate by providing an efficient means of reusing payload simulation software. The SRS was then duplicated so as to have isolated systems for unclassified and classified usage.

Payloads Intelligent Controller

The system architecture of the SMTF has evolved during the last decade and a thorough upgrade is in progress. The original configuration of the SVS included a Univac* 1100/44 host

*Later known as Sperry and now called Unisys.

computer with Interdata** 8/32 Intelligent Controllers (ICs) which were essentially smart interfaces between the host and the simulation equipment.

An 8/32 was added to each SVS base for payloads simulation because the host did not have the required computational capacity. The additional 8/32, called the Payloads Intelligent Controller (PLD IC), was interfaced to the SVS via shared memory. A Telemetry Interface Processor was attached to the PLD IC to support communications with the SCF during "joint-integrated" simulation.

PLS Computer System

In 1984 NASA started planning for simulation of the Centaur booster in the SMTF. The Shuttle flight schedule had a May 1986 launch date for the first Centaur mission and the SMTF had to be ready for Centaur training a few months before that.

The Centaur booster is a liquid fueled upper stage that was modified for stowage in the payload bay of the Space Shuttle. Centaur offered significant performance improvements over the IUS and was selected for use with payloads requiring greater thrust. NASA had plans to use it for many missions, the first for sending the Galileo spacecraft to Jupiter and the second for sending the Ulysses spacecraft around Jupiter to swing it out of the planets' orbital plane and then over the poles of the Sun[7].

Unlike other payloads carriers, the Centaur was liquid fueled which made it much more difficult to simulate. The propellants had to be transferred; tanks had to be vented, purged, and dumped (in abort sequences). Many variables had to be modeled along with the functions of the on-board computers that initiated, controlled, and monitored the sequence of events.

The PLD IC did not have adequate capacity for simulating Centaur. Estimates for the necessary computing power ranged between 2.6 and 6 times the power of the PLD IC[8]. Therefore, an upgrade was necessary and various upgrade options were considered[9]. The factors affecting the choice included hardware cost, software conversion cost, cost of new software, implementation schedule, growth path for the future, hours of training available, and security considerations.

In view of all the above factors, a Perkin-Elmer 3200 MPS was selected to replace the PLD IC and was named the Payloads Simulator (PLS). Unlike the ICs, this computer system is a multiprocessor consisting of one Central Processing Unit and as many as nine Auxiliary Processing Units.

Two operating systems were considered for the PLS: (1) OS/32, the vendor's commercial off-the-shelf operating system; and (2) the locally developed and maintained Real-Time Monitor (RTM) that was used on the 8/32s in the SMTF. Since RTM ran only on uniprocessor machines, the alternatives were to (1) enhance RTM to run on a multiprocessor system, or (2) adapt and augment OS/32 for use in real-time flight simulation. A study[16] recommended the latter. Accordingly, OS/32 was selected and real-time support software was developed for it. Payloads models that were written in Fortran did not have to be converted. However, assembler code that included operating system calls had to be converted.

Spacelab Simulator

The Spacelab is an exceptional payload in that it is the only one that carries people. Therefore, it is treated differently in the SMTF. A separate Spacelab Simulator (SLS) was built prior to any other payload simulation in the SMTF. The reason was to provide stand-alone Spacelab crew training independent of Shuttle crew training. Another reason for a separate SLS was security. Since Spacelab missions include foreign astronauts, the SLS must be capable of being isolated from the Shuttle simulators which are used to train for DOD classified missions.

There are two other simulators for training Spacelab crews, one in the Payload Crew Training Complex at Marshall Space Flight Center and one in West Germany[10]. However, the SLS at the JSC is the only one that operates in unison with a Space Shuttle simulator and can participate in an "integrated" simulation or a "joint-integrated" simulation with the MCC and a POCC. Also, the SLS can operate in a stand-alone mode.

As a stand-alone simulator, the SLS has its own crew station and instructor/operator station. The SLS crew station is not physically connected to the FB or MB crew stations in the manner in which the Spacelab is connected to

**Later known as Perkin-Elmer and now as Concurrent Computer Corporation.

the mid-deck of the Shuttle. This is because the narrow passage between the Orbiter cabin and the Spacelab module is not suitable for use in a 1g environment.

The Spacelab can be flown in various configurations. The manned module of the Spacelab has two configurations, a short one and a long one. In addition to the module, there can be as many as five pallets which hold experimental apparatus that is exposed to space. The Spacelab can be flown without a module, in which case an "igloo" containing the necessary subsystems takes the place of the module. So far there have been four Spacelab missions and all of them, except for the third mission, had a configuration consisting of a long module and a pallet. The configuration on the third Spacelab mission consisted of an igloo and 3 pallets. It had no habitable module. Many more Spacelab missions are planned during the next decade[11].

The SLS crew station represents the long module configuration. It includes replicas of the CRT (called the Data Display Unit), the keyboard, and the equipment racks in the module. The pallets and igloo are simulated in software. Interfaces between the Orbiter and the Spacelab, including power, thermal and voice communication, are simulated. However, the SLS does not include simulation of the scientific experiments.

The computers supporting the SLS are two 8/32s. The Spacelab's Command and Data Management Subsystem, which includes three on-board computers, is simulated within one of the two 8/32s. The other 8/32 simulates other on-board systems and also interfaces to the crew station and the instructor/operator station. The two 8/32s are coupled by means of shared memory. In addition, they can access the shared memory of either the FB or the MB simulator. This connection to shared memory is the coupling between the SLS and the SVS. The connection is switchable, for stand-alone operation of the SLS and for isolation during classified operations on the SVS.

Upgrading the SMTF

In 1984 and 1985 NASA and its contractors performed studies for upgrading the entire SMTF. The study team recommended an architecture[12] that would overcome the problems of obsolescence, capacity, reliability and maintainability. The study addressed deficiencies in the payload simulation area. Among them was the lack of a stand-alone payload simulation capability for checkout of payloads simulation and for part-task training of payload operations. Another deficiency cited was the inability to support mixed payloads, i.e. a short Spacelab configuration with other payload(s) in the Shuttle. The study recommended hosting the SVS and the PLS with one large computer system which might be a uniprocessor or a tightly coupled multiprocessor.

Due to budget and facility constraints and the need to provide ongoing training, the upgrade is being performed in four steps spread over several years. Step I, which was completed in June 1988, replaced the old 1100/44 host computers by new larger computers, the 1100/92, and replaced the four ICs with one Concurrent 3280 which is called the Base IC.

Step I of the upgrade did not directly affect the simulation of payloads but Step II will. One of the four projects in Step II is an upgrade of the SLS which will provide more computing power and thereby provide the opportunity to enhance the fidelity of the simulation. In addition, the upgrade of the SLS will bring it into a common computational environment with the rest of the payload simulation. This will permit the use of shared development resources and will alleviate the shortage of SLS development resources. This and other factors such as common configuration management will reduce software life-cycle costs.

The plans for the future include an OMV simulator for training astronauts and ground controllers in the MCC who will remotely pilot the vehicle. The OMV simulator may share computational facilities with other simulators in the SMTF.

Concluding Remarks

At the beginning of the Shuttle program, the SMTF was built for vehicle simulation only. However, as the program entered the operational era, there was a need to provide payload simulations of increasing complexity. To satisfy this demand, additional computing power was added.

Payload simulation in the SMTF is expected to grow in complexity. Higher flight rates will require efficient and rapid development of simulation software. The upgrade will provide the tools and facilities necessary for this.

Acknowledgement

This work was sponsored by NASA contract number NAS9-18057.

References

1. Usher, T. H. and Doetsch, K. H., "An Overview of the Shuttle Remote Manipulator System", Space Shuttle Technical Conference, NASA Conference Publication 2342, Scientific and Technical Information Branch, NASA Johnson Space Center, Houston, TX, 1983.

2. Kaplan, M. H., "Space Shuttle - America's Wings to the Future", Aero Publishers, Inc, Fallbrook, CA, 1978.

3. Hajare, A. R. and Brown, P. M., "A Computer Systems Upgrade for the Shuttle Mission Training Facility", AIAA Flight Simulation Technologies Conference, American Institute of Aeronautics and Astronautics, Washington, DC, 1988.

4. Williams, L. G. and Matthews, R. E., "Capabilities and Limitations of the Shuttle for Future Cargo Programs", Proceedings of the Nineteenth Space Congress, Canaveral Council of Technical Societies, Cocoa Beach, FL, 1982.

5. Stowman, J. A., Wagner-Bartak, C. G., and Doetsch, K. H., "A Real-Time Simulation Facility for the Development of Manipulator Systems with Man-in-the-Loop", 10th Space Simulation Conference, American Institute of Aeronautics and Astronautics, New York, NY, 1978.

6. NASA, "Space Transportation System User Handbook", NASA Scientific and Technical Information Office, Washington, DC, 1982.

7. Niesley, J. E., "Centaur for the 1980s", Proceedings of the Eighteenth Space Congress, Canaveral Council of Technical Societies, Cocoa Beach, FL, 1981.

8. Brown, P. M. , "Centaur Simulation Options Using the SLS Computers", JSC # 20059, The MITRE Corporation, Houston, TX, 1984

9. Mitchell, R. C. "Payload Simulator Upgrade: Considerations on the Options", JSC #19695, The MITRE Corporation, Houston, TX, 1984.

10. Shapland, D. and Rycroft, M., "Spacelab - Research in Earth Orbit", Cambridge University Press, Cambridge, England, 1984.

11. Lord, D. R., "Spacelab - An International Success Story", NASA SP-487, NASA Scientific and Technical Information Division, Washington, DC, 1987.

12. Hajare, A. R. and Brown, P. M., "Upgrading the Shuttle Mission Simulators", Simulators V, Simulation Series, Vol. 19, No. 4, The Society for Computer Simulation, San Diego, CA, 1988.

List of Abbreviations

AOS	Acquisition of Signal
CCTV	Closed Circuit Television
CRT	Cathode Ray Tube
DOD	Department of Defense
DPS	Data Processing System
FB	Fixed Base
GNS	Guidance and Navigation Simulator
GPC	General Purpose Computer
GPM	Generalized Payload Model
IC	Intelligent Controller
IOS	Instructor Operator Station
IUS	Inertial Upper Stage
JSC	(Lyndon B.) Johnson Space Center
LOS	Loss of Signal
MB	Motion Base
MCC	Mission Control Center
MCIU	Manipulator Controller Interface Unit
NASA	National Aeronautics and Space Administration
NSS	Network Simulation System
OMS	Orbital Maneuvering System
PAM	Payload Assist Module
PLD	Payloads
PLS	Payload Simulator
POCC	Payload Operations Control Center
RHC	Rotational Hand Controller
RMS	Remote Manipulator System
RTM	Real-Time Monitor
SCE	Signal Conversion Equipment
SCF	Satellite Control Facility
SID	Simulation Interface Device
SMTF	Shuttle Mission Training Facility
SLS	Spacelab Simulator
SRS	Simulator Reconfiguration System
SVS	Shuttle Vehicle Simulator
TDRSS	Tracking and Data Relay Satellite System
THC	Translational Hand Controller

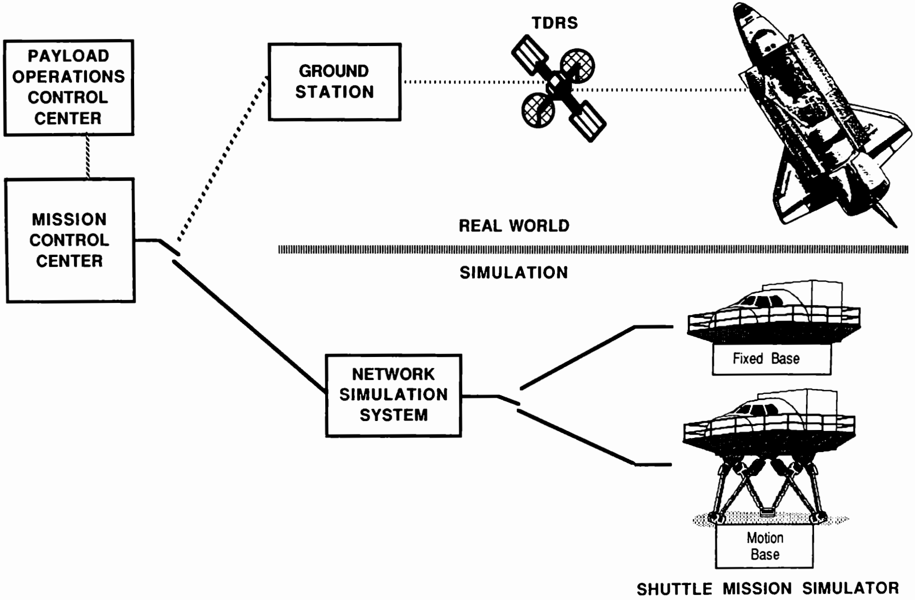

PAYLOAD OPERATIONS CONTROL CENTER
MISSION CONTROL CENTER
GROUND STATION
TDRS
REAL WORLD
SIMULATION
NETWORK SIMULATION SYSTEM
Fixed Base
Motion Base
SHUTTLE MISSION SIMULATOR

Figure 1
Joint-Integrated Simulation

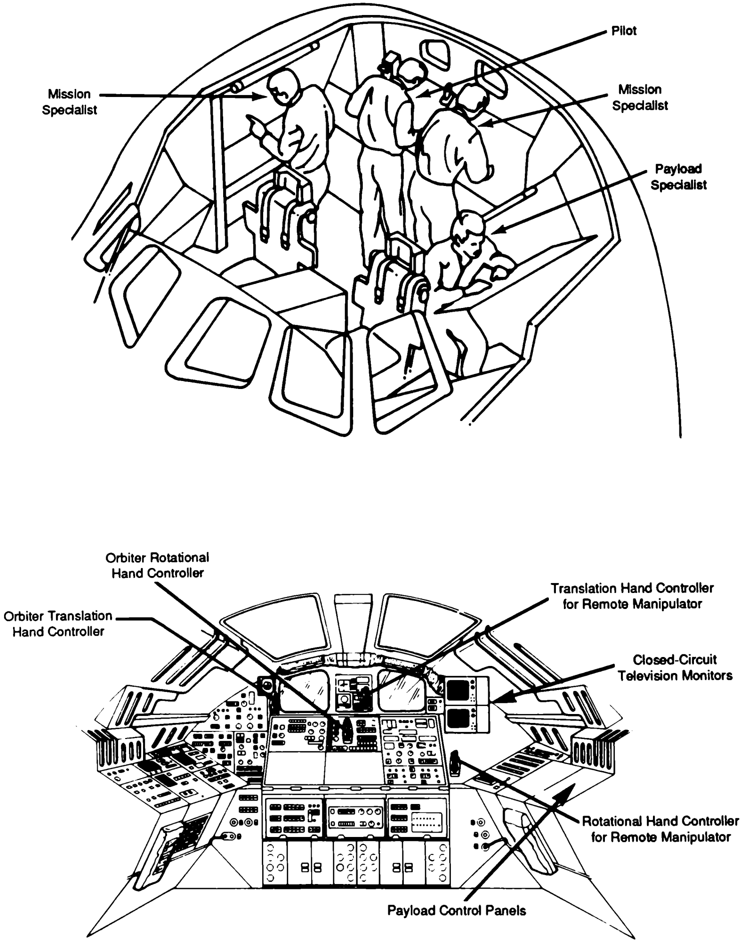

Pilot
Mission Specialist
Mission Specialist
Payload Specialist
Orbiter Rotational Hand Controller
Orbiter Translation Hand Controller
Translation Hand Controller for Remote Manipulator
Closed-Circuit Television Monitors
Rotational Hand Controller for Remote Manipulator
Payload Control Panels

FIGURE 2
AFT FLIGHT DECK

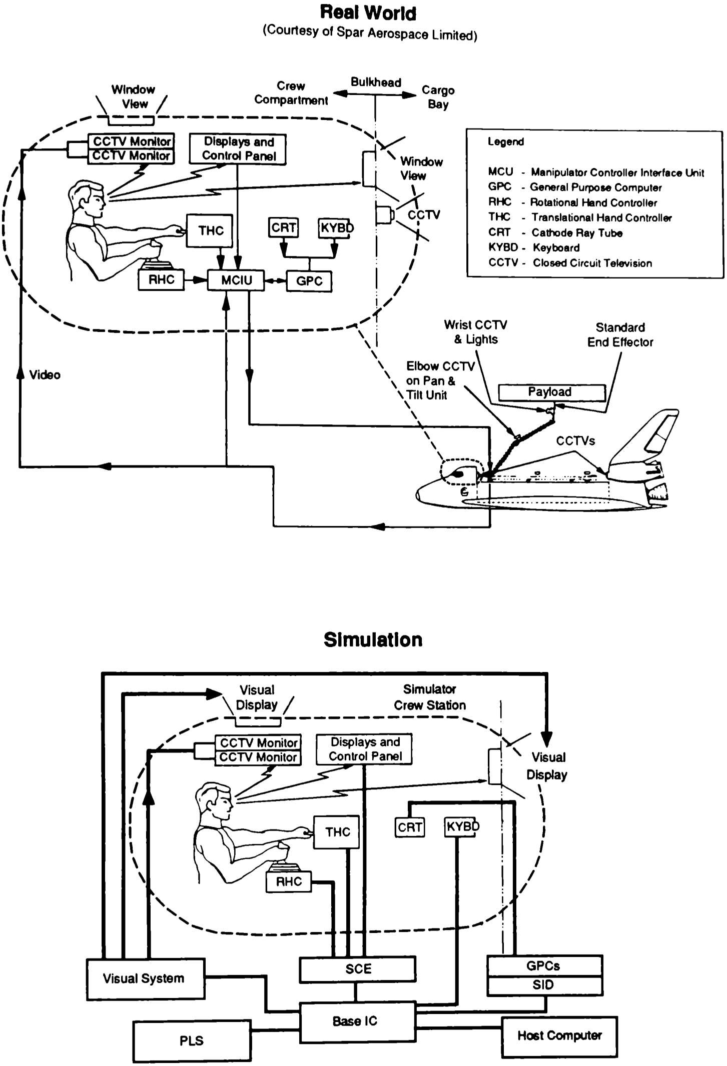


FIGURE 3
REMOTE MANIPULATOR SYSTEM SIMULATION

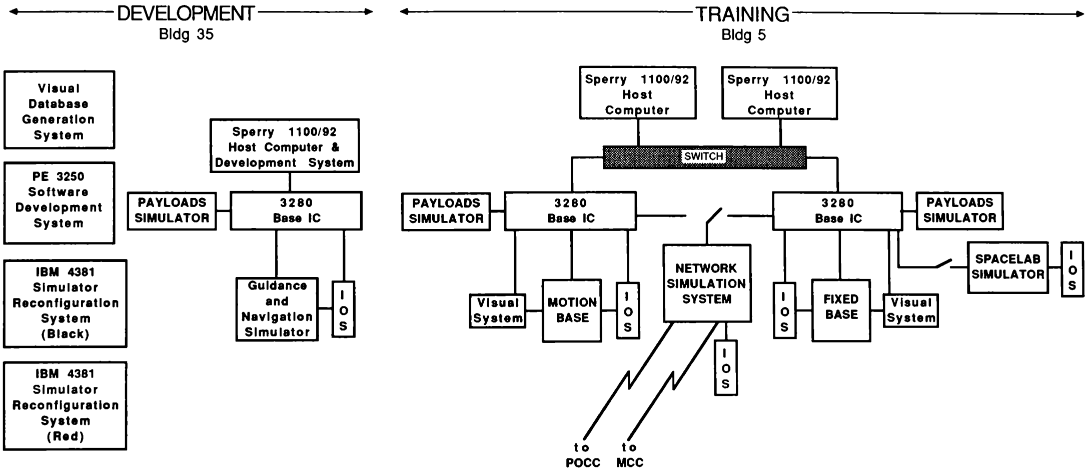

DEVELOPMENT
Bldg 35
TRAINING
Bldg 5
Visual Database Generation System
PE 3250 Software Development System
IBM 4381 Simulator Reconfiguration System (Black)
IBM 4381 Simulator Reconfiguration System (Red)
Sperry 1100/92 Host Computer & Development System
PAYLOADS SIMULATOR
3280 Base IC
Guidance and Navigation Simulator
I O S
Sperry 1100/92 Host Computer
Sperry 1100/92 Host Computer
SWITCH
PAYLOADS SIMULATOR
3280 Base IC
3280 Base IC
PAYLOADS SIMULATOR
NETWORK SIMULATION SYSTEM
SPACELAB SIMULATOR
I O S
Visual System
MOTION BASE
I O S
I O S
FIXED BASE
Visual System
I O S
to POCC
to MCC

Figure 4
Shuttle Mission Training Facility Configuration

89-3285-CP

THE FLIGHTS BEFORE THE FLIGHT:

AN OVERVIEW OF SHUTTLE ASTRONAUT TRAINING

John T. Sims*
NASA/Johnson Space Center
Houston, Texas

Michael R. Sterling **
Rockwell Shuttle Operations Company
Houston, Texas

ABSTRACT

Space shuttle astronaut training is centered at NASA's Johnson Space Center in Houston, Texas. Each astronaut receives many different types of training from many sources. This training includes simulator training in the Shuttle Mission Simulator, in-flight simulator training in the Shuttle Training Aircraft, Extravehicular Activity training in the Weightless Environment Training Facility and a variety of lectures and briefings. Once the training program is completed each shuttle flight crew is well-prepared to perform the normal operations required for their flight and deal with any shuttle system malfunctions that might occur.

INTRODUCTION

The United State's space shuttle is a unique vehicle with a unique mission. The crews who fly and operate this vehicle are also unique in the world of aerospace flight training. These men and women enter the program from many diverse backgrounds and with widely varying degrees of technical and professional experience. The transformation of these pilots, engineers, doctors, and scientists, into ready-to-fly space shuttle astronauts is a constant ongoing process centered at NASA's Johnson Space Center in Houston, Texas. This process involves many trainers, simulators, briefings, and lectures in Houston and at various points across the United States. Ensuring that each astronaut is ready for his or her flight is the responsibility of the Training Division of Johnson Space Center's Mission Operations Directorate.

ASTRONAUT CANDIDATE BACKGROUNDS

One of the unique problems encountered in shuttle astronaut training is the widely varying backgrounds of the trainees. Unlike most military and airline trainees who enter their respective training programs with similar backgrounds and experience, astronaut trainees, known as astronaut candidates or ASCANs, enter the training program with a variety of different backgrounds and widely varying degrees of technical and professional experience. There are two types of shuttle astronauts, thus two types of trainees, pilots and mission specialists. Pilot astronauts serve as both shuttle commanders and shuttle pilots. During flight the commander has the onboard responsibility for the vehicle, crew, mission success and safety of flight. The pilot assists the commander in controlling and operating the vehicle. In addition, both the commander and the pilot may assist in payload operations and other orbit activities. Mission specialist astronauts have overall responsibility for the coordination of shuttle operations in the areas of experiment and payload operations. They are required to have a detailed knowledge of shuttle systems, as well as detailed knowledge of the operational characteristics, mission requirements and objectives, and supporting systems and equipment for each payload element on their assigned missions. Mission specialists will perform Extravehicular Activities (EVA), payload handling using the Remote Maniupulator System (RMS), and perform specific experiment operations. In addition one mission specialist for each flight serves as "flight engineer" and assists the commander and pilot in ascent and entry operations.

The basic requirements for pilot and mission specialist astronaut are at least a bachelor's degree in engineering, biological science, physical science, or mathematics and related experience, the ability to pass a NASA space physical, and to meet certain height restrictions. In addition, it is required that pilot applicants have at least 1,000 hours pilot-in-command time in jet aircraft and flight test experience is highly desirable. While all ASCANs meet these minimum requirements their actual backgrounds and experience are usually much greater and vary widely. For example one pilot was a civilian test pilot with over 7,000 hours in jets and helicopters and degrees in aeronautical engineering and business. Another pilot has an aerospace engineering degree and 2,000 hours of flight time. Still another has degrees in electrical science and systems management and 3,400 hours of flying time. Mission specialist backgrounds vary even more widely. One mission specialist has a degree in mechanical engineering and a doctorate of medicine as well as 1,000 hours of flying time. Another has a doctorate in astronomy and astrophysics and no flight experience prior to joining NASA. Another has degrees in biological sciences and microbal ecology and a doctorate in civil and environmental engineering. Still another has a degree in aerospace engineering and over 4,400 hours of flying time.

* Shuttle Mission Simulator Team Lead
Member AIAA

** Shuttle Mission Simulator Control/
Propulsion Instructor
Member AIAA

It can be seen from this varied mix of trainees that the training program must be thorough yet very flexible to accommodate them all. It must simultaneously be able to fill in the gaps in their backgrounds while not being overly boring and dull when covering subjects that the trainee may very well be a leading expert. Most importantly the training program must assure that all the trainees achieve a very high level of shuttle operations and systems knowledge. The program is aided significantly by the fact that the varied mix of trainees all have one thing in common. They are all very highly motivated to learn and actively participate in the training process.

INITIAL ORIENTATION

When first named to the Astronaut corps, an ASCAN will find him or herself facing a barrage of basic, orientation tasks. This consists of many things, ranging from aircraft operations instruction, to Space Shuttle specific lectures, to trips to various NASA centers. Even though many of these tasks may not directly pertain to flight training, the shear time involved is a flight training issue.

Only about 50% of those chosen as Astronaut Candidates enter the space program with experience in high performance aircraft. Those that do enter the program with this experience find themselves isolated from military flying units. In the interest of gaining or maintaining this experience, NASA operates 25 T-38A jet training aircraft. These aircraft, housed and maintained at Ellington Field in Houston, Texas, perform three major functions. First, they allow pilot astronauts to maintain their proficiency at flying high performance aircraft. Second, they allow astronauts to travel from their home base in Houston to locations across the country as the job requires. Third, they allow the pilot astronauts to practice Space Shuttle approaches. Flying the T-38 in an aerodynamically dirty configuration allows the pilots to approximate the 19 degree glideslope approach of the Space Shuttle and affords them the most readily available tool to practice this task.

For many of the ASCANs, T-38 training is little more than a refamiliarization with an aircraft from their past. Others enter the program with experience in rotary wing aircraft. These ASCANs are typically given fixed wing flight training and soon gain T-38 qualification. Still others enter the program with little or no flight experience at all. These ASCANs take ground school and are familiarized with T-38 systems operations.

NASA is a multi-faceted, multi-talented entity that exists in a multitude of locations and performs a multitude of tasks. Though each individual NASA center performs a different function and each deals with a different field of aerospace investigation, all of them deal with the Space Shuttle program in some way. No doubt, at some point in an astronaut's career, he or she will rely on people from NASA centers other than the Johnson Space Center (JSC) for information. In order to build lines of contact and to promote the "team" nature of the Space Shuttle program, the ASCANs embark on a tour of both NASA centers and the factories where Space Shuttle parts are made. Although this may sound trivial, this task occupies a great deal of their time early in the program.

As in any flight training program, one must always maintain safety as the foremost priority. This is no less true of the Space Shuttle program. The ASCANs spend a great deal of time participating in survival oriented training courses. These courses require quite a bit of time and travel. They include: physiological training at JSC, land survival training at Fairchild AFB, water survival training and parasail training Homestead AFB, and SCUBA training/certification at JSC.

In addition to the T-38 training mentioned earlier, all ASCANs are required to go through a KC-135 familiarization flight. The modified KC-135, affectionately known as the "Vomit Comet", is used to give participants a short term exposure to weightlessness. It does this by flying a series of parabolas. After fifteen or twenty of these, the airplane's nickname suddenly makes sense.

Pilot astronauts also attend another T-38 ground school, designed to train them to fly the Space Shuttle approaches mentioned earlier. These approaches, called L/D's for Lift over Drag (though they involve little L and a lot of D), are taught in Houston by NASA pilots. The use of T-38's to fly L/D's not only allows relatively easy access to Space Shuttle approach training methods, it is also used as a stepping stone to a higher fidelity approach and landing trainer called the Shuttle Training Aircraft (STA).

The last group of "orientation" tasks that the ASCANs participate in deals with topics that may be considered to be uniquely space-related. They spend this time being fitted for clothing, testing space food, being trained in camera use, and public relations.

At the same time that the ASCANs are involved in the aircraft orientation, survival training, and NASA center tours they attend a series of basic lectures on Space Shuttle systems. This lecture series is where the ASCANs first encounter the most unique aspect of training for space flight, the source of the training. They depend on personnel from the Engineering Directorate at JSC for highly technical information concerning Shuttle systems. They depend on personnel from the Mission Operations Directorate at JSC for the practical, operational information and training needed for space flight and they depend on astronauts who have flown for information on what space flight is really like. There is no one group to depend on for all information. The topics of these lectures vary as well. They range from the highly technical, like a lecture on the intricacies of the Main Propulsion System, to the operationally detailed, like a lecture entitled Ascent Overview.

PERSONAL STUDY AND SYSTEMS TRAINING

Once the ASCANs have completed their initial orientations and briefings they begin a phase of personal study and a series of classes in the Single Systems Trainer (SST). The ASCANs begin to read workbooks on various shuttle systems, flight phases, and abort modes. These workbooks have been written by MOD Training Division instructors with help and input from various other MOD personnel. Each workbook is a stand-alone reference and information book describing a particular subject in detail. Shuttle systems workbooks deal with individual orbiter systems such as the Auxiliary Power Units (APUs), the Reaction Control System (RCS), or the Data Processing System (DPS). Some shuttle system workbooks describe groups of related system or instruments such as Guidance and Control Sensors or Mechanical Systems. There are workbooks describing various flight phases such as ascent and entry, as well as specific subjects in each phase such as Landing and Rollout, Ascent Targeting, and Rendezvous and Proximity Operations. There are also several workbooks dealing with shuttle ascent abort modes. These workbooks describe both the flight profile of these aborts as well as specific abort procedures.

Concurrent with studying the training workbooks, the ASCANs begin to take classes in the Single Systems Trainer (SST). The SST is a low fidelity mock-up of the orbiter flight deck with various shuttle systems simulated with math models. It is designed to train one shuttle system at a time, hence the name Single Systems Trainer. SST classes are taught by one instructor to usually one or two students at a time. The typical class is two hours in length and in this time period the instructor and student review and reinforce the student's personal workbook study on one particular shuttle system. There are three types of SST classes for most systems. These are basic system operations classes, basic system malfunction classes, and advanced system review classes. In the operations classes the students review the nominal operating characteristics and procedures for the particular shuttle system being studied. They exercise these procedures and observe the system performance through all flight phases. In the system malfunction classes the various malfunctions and corrective procedures for the system under study are reviewed and exercised. The instructor will introduce a malfunction into the system model and the student will exercise the appropriate procedure in response. The system review classes are designed as refresher courses for students who have taken the operation and malfunction classes and wish to keep up their proficiency in that particular system. Individual review classes are tailored to individual student needs and desires and can cover any aspect of the system under review.

SHUTTLE TRAINING AIRCRAFT

Simultaneously with SST training, pilot ASCANs begin specialized training in shuttle flying techniques. It is easy to see that NASA faces a fairly specialized task in trying to train astronauts to perform the ascent phase of a Space Shuttle flight. It is less obvious that they faced similar problems in deciding on a means by which to train astronauts to land the Space Shuttle. It is true that all of the pilot astronauts have a great deal of experience landing various types of aircraft. However, none of them have much experience at landing a 200,000 pound vehicle that touches down at an airspeed of 197 knots and at a sink rate of two to three feet per second. In training for this task, NASA employs a number of different methods and aircraft. It has already been mentioned that NASA T-38's are used in an aerodynamically dirty configuration to roughly demonstrate shuttle approaches, but the majority of the responsibility for training pilot astronauts to land the Space Shuttle lies squarely on the shoulders of an airplane that NASA calls the Shuttle Training Aircraft (STA).

The STA is a highly modified Grumman Gulfstream G-2 that is forced to do things that can only be described as an aircraft structures specialist's nightmare. The basic G-2 airframe has been reinforced its factory supplied Fowler type flaps have been replaced with board type elevons. Internally, the STA only vaguely resembles a stock G-2. Its usual control system has been replaced with one which allows the STA to closely approximate the handling qualities of the Space Shuttle, and most of the cockpit instrumentation and controls have been replaced with shuttle equipment. At an altitude of approximately 35,000 feet, the STA's main landing gear are deployed (main gear only, the nose gear remains stowed) and its engines are placed at flight idle. The thrust reversers are then deployed and the engines are slowly returned to full thrust. It is not difficult to imagine the effect this type of use must have on the structures of the STA. The thermal cycling on the thrust reversers alone, due to their deployment at the cold temperatures at 35,000 feet followed by the heating from engine exhaust, is enough for concern. As might be expected, the STA is one of the NASA's most often inspected aircraft.

As in most flight training programs, a pilot astronaut can expect to spend some time in an aircraft simulator before he is exposed to the actual aircraft. STA training is no different. Before being allowed to fly the STA, pilot astronauts must first complete qualification runs in the Shuttle Mission Simulator at Johnson Space Center. These qualification sessions are supervised by the instructor pilots who will eventually train the ASCANs in the STA itself.

After completion of their qualification sessions, the ASCANs are placed on an extensive

STA training schedule. The schedule consists of twenty trips to a number of different shuttle landing sites where the training will take place. After arriving at the site, the pilot astronaut will fly between ten and thirteen approaches to the point of simulated vehicle touchdown. There are two problems with STA training caused by the gross size difference between the Gulfstream G2 and the Space Shuttle. When the shuttle's main landing gear touch the runway surface, the commander's eye level is still about thirty feet above the surface of the runway. This is considerably higher than the pilot's eye level at touchdown in the STA. Therefore, the STA is impractical as a rollout training device. The second problem has to do with atmospheric turbulence on approach. The smaller mass of the STA makes it far more susceptible to attitude and velocity changes due to turbulence and wind gusts. In spite of these problems, the STA remains NASA's most effective tool for training shuttle approaches. Completion of their scheduled twenty sessions does not end the pilot astronauts training in the STA. Each pilot is scheduled for STA sessions at least once per month to maintain proficiency. The frequency of the sessions increases as the pilot nears launch time.

VERTICAL MOTION SIMULATOR

Although the STA provides an excellent simulation of shuttle approach handling characteristics, this simulation ends at simulated main gear touchdown. It is impossible to accurately simulate the shuttle's rollout handling characteristics in the STA. For this NASA employs a high-fidelity motion simulator at NASA/Ames Research Center called the Vertical Motion Simulator (VMS).

The VMS consists of a crew cabin which moves horizontally along a thirty foot long beam. This beam may move up and down within a five story building. To augment the motion cues, the cabin may also be rocked and tilted using hydraulic actuators. This by far affords the most realistic shuttle rollout simulation. Although the Astronaut Office and NASA Training Division do not have scheduled access to the VMS for training purposes, JSC schedules approximately eight weeks every six months for engineering evaluation. Of that eight weeks, about 2 weeks are used for crew training. This training time is considered invaluable by all involved.

SHUTTLE MISSION SIMULATOR

The Shuttle Mission Simulator (SMS) is the primary training tool for shuttle astronaut training. It is the only high-fidelity simulator capable of training flight crews for all phases of a shuttle mission, from lift-off minus thirty minutes through touchdown and rollout. This includes prelaunch check-out, ascent, aborts, on-orbit operations, entry and landing. The SMS complex consists of a fixed base simulator (FB), a motion base simulator (MB), a Network Simulation System (NSS), a Spacelab Simulator (SLS), and supporting instructor stations, operator stations, and computer facilities.

The fixed base simulator consists of high-fidelity mockup of the shuttle orbiter flight deck and middeck. The motion base has only the orbiter flight deck. Each simulator is supported by a dedicated Sperry 1100/92 host computer and Concurrent 3280 **base intelligent** controller. Also each simulator base has a dedicated set of five shuttle General Purpose Computers (GPCs). These GPCs are nearly identical to those used in the real vehicle and they operate with the real flight software to be used for the particular mission being simulated. Each base is equipped with a visual system. This system provides computer generated images of the shuttle launch tower and Kennedy Space Center, various shuttle landing sites around the world including Edwards Air Force Base, Ben Guerir, Morocco, Moron, Spain, and Hawaii. It also provides images of starfields and on-orbit payload operations.

The motion base is similar to the fixed base except that it contains the forward flight deck only. It is equipped with a six-degree of freedom hydraulic motion system. This system simulates the movement felt by the crew during an actual shuttle ascent or entry. For launch simulations the motion base is pitched such that the crew are laying on their backs. This aids in simulating the G-loading felt during ascent.

The Network Simulation System is used to simulate the global shuttle tracking and communications system including the Tracking and Data Relay (TDRS) satellites. The NSS is used to link an SMS base to the Mission Control Center during integrated simulations involving flight controllers. the NSS simulates tracking data, voice and data downlink, and voice and commanding uplink. The NSS also simulates the loss and acquisition of signal as the shuttle orbits the earth.

The Spacelab Simulator is used to train Spacelab missions. It is a mock-up of the pressurized Spacelab module carried in the shuttle payload bay. It can be integrated with either SMS base and simulates Spacelab functions, and experiment operations.

The primary function of the SMS is, of course, shuttle astronaut training. However, the SMS facility has several other important functions. The first of these is crew procedure verification. When new shuttle procedures, both for nominal operations and malfunction situations, are written they are tried out in the SMS with a shuttle crew member to determine their accuracy and understandability. They are also evaluated as to whether the procedure could realistically be accomplished in a timely manner. Very often new procedures must be rewritten and tested several times before they are approved for flight. Another use of the SMS is for instructor familiarization. This time is used for instructors to train themselves and practice techniques and cases without a crew. The SMS is also used for shuttle mission support. Should a problem arise in an actual shuttle flight, the SMS can be brought on line to simulate the problem and allow various solutions to be tested on the ground before they are passed to the orbiting flight crew. Finally, the SMS is used for simulator software development. This time is used to develop new SMS software to support upcoming flights. For example, new payloads must be modeled and different flight profiles must be simulated. This time is also used for fixing problems or making changes in existing shuttle systems models.

Each training session on the SMS is supported by a team of instructors. The basic SMS team consists of five people, the team lead, the control/propulsion instructor, the systems instructor, the data processing/navigation instructor, and the communications/instrumentation instructor. The team lead is in charge of the SMS team. It is his or her job to coordinate the other instructors actions, ensure the training session is conducted in an orderly and timely manner, and that the training requirements for the lesson are met as defined in the Shuttle Crew Training Catalog. The control/propulsion instructor is responsible for training the shuttle's propulsion systems, such as the Reaction Control System and the Orbital Maneuvering System, and for training the shuttle's control systems, such as the Digital Auto Pilot and the Flight Control System. The control/propulsion instructor is also responsible for instructing in shuttle guidance and flight techniques. The systems instructor is responsible for training all the orbiter's mechanical, electrical and environmental systems, such as the Auxiliary Power Units, the Payload Bay Doors, and the Fuel Cells. The data processing system/ navigation instructor is reponsible for the training of the orbiter's General Purpose Computers and their associated data bases as well as the shuttle's navigation equipment. The Communications/instrumentation instructor is responsible for training in the orbiter's data and voice communications systems and the orbiter's data collection instrumentation system. One team of instructors is assigned to each shuttle mission. The assigned team supports every SMS session in which their crew participates. In addition to the above instructors there are several specialty instructors who support SMS training as needed. For each flight there are one or more payloads instructors assigned. These instructors are responsible for training the payloads on a particular mission. If the flight carries a Remote Manipulator System, i.e. the shuttle's robot arm, or if the flight involves a rendezvous, specialists in these areas are assigned. There are also specialty instructors in aborts and ascent/entry flight techniques.

"PILOT POOL" TRAINING

SMS training sessions are divided into two categories. These are "pilot pool" sessions and "mission specific" sessions. After an ASCAN completes the year long "basic training" program he or she graduates to "astronaut" status and is eligible for "pilot pool" training. This training is designed to accomplish two tasks. The first of these tasks is to improve the astronaut's understanding of Space Shuttle systems operations, and expose them to shuttle crew coordination. The second task of these sessions is to maintain proficiency in shuttle operation among the senior astronauts.

"Pilot pool" sessions are structured to allow less experienced astronauts to draw on the experiences of their seniors. This is accomplished with the help of an "instructor pilot" (IP). An IP is an astronaut who is assigned to participate in an SMS training session for the purpose of aiding in the instruction of the other astronauts involved in the session. The criteria for being assigned to an SMS training session as an IP vary depending on the flight rate at the time of the session. If the flight rate is low, there will be a ready supply of astronauts with flight experience available to act as IPs. When available, those with flight experience are preferred as IPs. If the flight rate is high and those with experience are training for flights, the IP is chosen based on seniority. During "pilot pool" training sessions, the senior astronaut is expected only to supplement the training of his less experienced colleagues. The responsibility for the training accomplished in these sessions belongs to the SMS training teams.

Due to mission length and facility availability, it is seldom if ever possible to simulate an entire mission at one time. To try to compensate for this, "pilot pool" training is roughly divided into several major types of training sessions, some of which deal with flight phases and some of which deal with systems demonstrations.

For example, during ascent/abort familiarization sessions, the crew is subjected to four hours of repeated ascents. The primary purpose of this type of training is to acquaint the astronauts with methods of dealing with both mechanical systems failures and computer problems during this very dynamic phase of flight. The most important type of failure covered in this type of session is that which causes an abort. All types of abort scenarios are demonstrated as often as possible. Astronauts are also given concentrated training on flying Space Shuttle entries with serious failures. This often includes scenarios which extend from deorbit all the way through landing and rollout. Another type of training session deals with specific system training. This type of training is often also tied to a particular flight phase. Classes which cover such topics as how the Backup Flight Software influences shuttle ascents or how shuttle entry guidance works are common. There are also several classes dealing with common shuttle payloads.

MISSION SPECIFIC TRAINING

Once a group of astronauts is assigned to a flight they begin mission specific training. Training usually starts approximately eight to nine months before the scheduled launch date.

Training begins with two weeks of review of basic shuttle systems in the Single Systems Trainers. Once this is complete the crew begins "flight-similar" training. "Flight-similar" training occurs at the SMS using simulator software from a previous shuttle flight that has a similar ascent/entry profile and if available, a similar payload. The simulator software includes not only the software required to model various shuttle systems but the software required to model the specific payload for that flight and that particular mission's flight software for the on-board general purpose computers. The simulator software is called a training load. "Flight-similar" training consists of ascent/abort training, orbiter systems and procedures training, orbit flight operations, and deorbit and entry training. Depending on the shuttle mission, there may also be rendezvous training and Remote Manipulator System training. If a similar payload is supported on an existing training load there will be "flight-similar" payload training as well. SMS training is usually scheduled in four-hour blocks. Each block is dedicated to a particular subject such as ascent/aborts. During a typical ascent session several different runs are made starting a few minutes before lift-off and ending on-orbit for an "uphill" run or after landing for an abort run. During each run each instructor inputs failures for his particular system based on a lesson plan, called a script, the instructors have written in preparation for the training session. The crew reacts to these systems failures and works the proper procedures in response. These failures often require an ascent abort such as a Return to Launch Site (RTLS) or Transatlantic Landing (TAL). The instructors monitor the crew's actions in response to the failures and advise and critique as necessary. The instructors also monitor the crew's performance of nominal shuttle procedures. Most other lessons follow the same general format.

The first part of the crew's "flight-similar" training is called stand-alone training. In stand-alone training the instructors act as mission control, advising the crew, making normal ground calls, and making abort calls. The real flight controllers are not involved in these lessons. Therefore the lessons concentrate on procedures and malfunctions that the flight crew can work with a minimum of ground information. Since this is "flight-similar" training with a training load from a different flight the emphasis of these integrated sessions is on solving systems problems and running generic timelines rather than completing specific mission objectives.

Once the crew has been training in a stand-alone mode for several weeks they begin to take "flight-similar" integrated sessions. In integrated sessions the flight controllers participate in the Mission Control Center (MCC) just as in a real flight. The Network Simulation System is brought on-line and transmits data from the SMS to the MCC simulating the real shuttle vehicle's telemetry stream. With this data the flight controllers can track the orbiter, devise solutions to problems, and advise the flight crew.

The flight crew continues "flight-similar" training until eleven weeks before their scheduled launch date. At this time their "flight-specific" training load is delivered. This training load is tailored to match their flight. It contains the flight's ascent profile, payloads and flight software. When this load is delivered training shifts into high gear. The crew spends twelve to sixteen hours or more in the SMS per week continuing both stand-alone and integrated training. Again there is the mix of ascent/abort training, orbiter systems and procedures training, orbit flight operations, and deorbit and entry training. However, there is a much greater emphasis on payload training and mission specific training such as rendezvous or Remote Manipulator training. The integrated sessions especially stress mission objectives, such as deploying payloads and exercising the mission timeline.

Once the "flight-specific" training load is delivered a series of joint integrated simulations maybe scheduled. These simulations involve the flight crew in the SMS, the flight controllers in the MCC, and the payload controllers at their control center. The payload control center is usually located away from JSC in Houston. Some payload control centers include the Jet Propulsion Laboratory in California, Marshall Space Flight Center in Alabama and Department of Defense sites. These joint simulations usually run for eight to sixteen hours and focus on communication between the flight crew and the ground and between the various control centers. A variety of payload malfunctions are worked and real-time timeline changes are practiced. For many flights a special joint integrated simulation is scheduled. This is a Long Duration simulation that usually last anywhere from 32 to 36 hours. During the "Long Sim" the entire flight is practiced beginning at lift-off. The simulator and control centers are manned 24 hours a day just as in the real flight. Again the emphasis is on flight-specific mission objectives and payload operations.

When not involved in simulator training the flight crew must attend an extensive list of briefings concerning their mission. They must absorb information from hours of highly detailed briefings on the nature of the payloads to be carried on their flight. This not only includes descriptions of the payloads, histories of the payloads, descriptions of similar experiments, and expected outcomes, but also includes discussions of procedures to be used in dealing with the payload. Crew members are often required to fly to customer facilities for short stays to receive these briefings and the training required to correctly perform the activities required by the payload. The crew also received briefings on the peculiarities of their mission. For example, they are briefed on any differences in their propellant management, any changes to their flight software, and any new orbiter procedure changes.

The crew is also briefed on any Detailed Test Objectives (DTOs) that they are expected to accomplish. The Space Shuttle is still an experimental vehicle. As such, there are still some gaps in its flight data. Each flight has a list of pre-determined flight test activities to be accomplished to fill these gaps.

SPECIALTY TRAINING

In addition to all the flight crew's other training there is a significant amount of specialty training to be accomplished. Specialty training is given to crews based on their mission requirements. One such area of training is that of Extravehicular Activity (EVA). Every crew goes through a certain amount of EVA training for every flight. There are three types of EVA, planned, contingency, and emergency. Planned EVAs are those where it was scheduled in advance of the flight that two crew member's would go outside the orbiter to perform some function such as work on a payload or a retrieved satellite. Contingency EVAs are those where crew members have to go outside to perform some function critical to mission success such as manually aid in a payload deployment. Contingency EVAs are planned in advance as a response to a possible payload malfunction, but are only executed should a malfunction occur. Emergency EVAs are those where a crew member must go outside to perform a function critical to orbiter safety, such as manually closing the payload bay doors. All flight crews are trained in Emergency EVA procedures. However, planned or contingency EVA training is dependent on the mission requirements. EVA training includes briefings and lectures on procedures and equipment. There is also an Extravehicular Mobility Unit (EMU), or space suit, malfunction single systems trainer. Most of the EVA training, however, occurs at the Weightless Environment Training Facility (WETF). The WETF is a large, thirty foot deep water pool that contains a mock-up of the shuttle's payload bay. It can also be configured to contain mock-ups of various shuttle payloads. The crewmember in training wears a special EMU, or space suit, configured for water work and weighted so that it will have neutral buoyancy when submerged. In this way weightlessness is simulated and the crew member can practice a variety of EVA tasks.

Another area of specialty training is that of the Remote Manipulator System (RMS). As has been mentioned before RMS training can occur at the SMS. RMS training at the SMS is high-fidelity since various other orbiter system failures which impact the RMS can be demonstrated. Also the affect of RMS activities in conjunction with other mission timeline activities is also trained. However, there are other facilities involved in RMS training also. One of these is the One-G trainer. This is a full scale mock-up of the orbiter flight deck and payload bay. An RMS mock-up is then used to practice berthing and unberthing various payloads. Other RMS activities can be trained here as well. The Shuttle Engineering Simulator (SES) is also used for RMS training. This simulator's mathematical model of the RMS is fully dynamic, i.e. takes into account the mass and inertia of the RMS and an attached payload. This capability does not exist in the SMS RMS model.

Rendezvous and Proximity Operations are another specialty area of training. All flight crews get a limited amount of rendezvous/proximity operations training. This training is to cover an event where an EVA crew member (whether planned or otherwise) has somehow drifted away from the orbiter and needs to be rescued. On flights where a rendezvous is part of the flight plan, satellite repair or retrieval for example, there is an extensive amount of training in rendezvous in the SMS and the SES. This training is done both stand-alone and integrated with the mission control center. Nominal rendezvous profiles are exercised as well as contingency profiles brought about by low propellant quantities or off-nominal orbits.

Finally, all crews get a certain amount of specialty training in the One-G Trainer on miscellaneous orbiter operations. These include galley operations, use of the Waste Collection System (the orbiter toilet), and routine orbiter housekeeping chores. There is also training in in-flight maintenance. In-flight maintenance can be performed on certain flight critical systems on-orbit should a failure occur. The failed component is removed and replaced with a spare or switched with a component from a non-critical location.

CONCLUSION

Training at JSC continues for flight crews until three days before launch. At this point the flights before the flight end and the crew departs for the Kennedy Space Center for the real thing. Training individuals to be members of a shuttle crew is a long involved process including briefings, training manuals, and low and high-fidelity simulators. The training process is complicated by the different backgrounds and experience levels of astronaut candidates as well as the unique challenges in manned spaceflight training. However, once the training program is completed each crew is well-prepared to perform the normal operations required for their flight in a timely manner and deal with any shuttle system malfunctions that might occur.

89-3268-CP

SUMMARY OF PROCEEDINGS OF THE FIRST MEETING OF THE NASA AMES SIMULATOR SICKNESS STEERING COMMITTEE

Lawrence J. Hettinger, Michael E. McCauley

Monterey Technologies, Inc.
P.O. Box 223699
Carmel, California 93922

Anthony E. Cook

Flight Systems and Simulation Research Division
NASA Ames Research Center
Moffett Field, California 94035-1099

James W. Voorhees

Crew Station Research and Development Branch
Army Aeroflightdynamics Directorate
Moffett Field, California 94035-1099

Abstract

A program of research to investigate simulator induced sickness has recently been initiated under the sponsorship of NASA Ames Research Center at the request of the U.S. Army. A major goal of the program is to coordinate the diverse efforts that have been made by various branches of the Armed Forces to investigate and eventually eliminate the problem of simulator sickness. As part of this program, a Simulator Sickness Steering Committee has been assembled, comprised of eighteen representatives from the Army, Air Force, Navy, NASA, NATO, academia and industry. The proceedings of the first meeting of the NASA Ames Simulator Sickness Steering Committee will be summarized and discussed.

Background

This paper summarizes the first meeting of the NASA Ames Simulator Sickness Steering Committee, held at NASA Ames Research Center on September 27-29, 1988. The major objectives of the meeting were to: 1) provide a general overview of the topic through the presentation of position statements by committee members; 2) achieve a consensus on the implications, causes, and recommended approaches to finding solutions; and 3) develop a working plan for future meetings and activities of the Steering Committee.

The NASA Ames Simulator Sickness Steering Committee was conceived by Anthony Cook, NASA Ames Flight Systems and Simulation Research Division, to provide a common ground for representatives of the U.S.

Armed Forces, NASA, and NATO to exchange information and to promote a better understanding of the simulator sickness syndrome. The objectives of the Steering Committee are to:

(1) facilitate the exchange of information about simulator sickness among the U.S. Armed Forces, NATO, and other organizations;

(2) recommend standardized and innovative methods for assessing the incidence and severity of simulator sickness;

(3) identify and assign priorities to research issues;

(4) encourage the development of simulation engineering design criteria to reduce the problem;

(5) foster the development of guidelines for simulator usage, calibration, and maintenance to reduce the problem;

(6) identify and promote other approaches to understanding and controlling simulator sickness.

The Steering Committee members are identified in Table 1. The members were selected to represent NASA, the three major branches to the Armed Forces, NATO, industry, and academia. In addition, members were selected to provide representation from a variety of disciplines, including engineering, physiology, psychology, and medicine. Individually, committee members possess a broad range of expertise in flight simulation, including training, design, acquisition, research and development, and the psychological and physiological processes involved in motion, space, and simulator sickness.

Simulator Sickness

Definition

A common definition of simulator sickness, although not necessarily endorsed by all members of the Committee, is as follows: Simulator sickness refers to the constellation of signs and symptoms of motion sickness and related perceptual after-effects that occurs in ground-based vehicular simulators. The simulator sickness syndrome is characterized by adverse symptomatology experienced either during or after exposure to simulated motion scenarios that would not produce sickness in the actual vehicle. Common symptoms are nausea, general discomfort, stomach awareness, and fatigue. Commonly observed signs include pallor, sweating, salivation, and postural instability. In rare cases, severe nausea, vomiting, visual flashbacks or delayed-onset of symptoms may occur.

Differing opinions concerning the prevalence and the indices of simulator sickness were reflected throughout the Steering Committee's discussions. For example, there was active

TABLE 1
MEMBERS OF THE NASA AMES SIMULATOR SICKNESS STEERING COMMITTEE

Anthony M. Cook (Convener)	NASA Ames Research Center
Michael E. McCauley (Chair)	Monterey Technologies, Inc.
Alan J. Benson	RAF Institute of Aviation Medicine
Richard S. Bray	NASA Ames Research Center
Frank M. Cardullo	State University of New York
Walter S. Chambers	Naval Training Systems Center
Malcolm M. Cohen	NASA Ames Research Center
Lawrence H. Frank	Pacific Missile Test Center
Hal Geltmacher	Air Force Human Resources Laboratory
Daniel W. Gower, Jr.	U.S. Army Aeromedical Research Laboratory
Fred E. Guedry, Jr.	Naval Aerospace Medical Research Laboratory
Robert S. Kennedy	Essex Corporation
L. Matthew Landry, Jr.	McDonnell Douglas Helicopter Company
Michael G. Lilienthal	Naval Training Systems Center
Grant B. McNaughton	March AFB
Kenneth E. Money	Defence and Civil Institute of Environmental Medicine
John B. Sinacori	John B. Sinacori Associates
James W. Voorhees	U.S. Army Crew Station Research and Development Branch

discussion among the Committee members as to whether vision-related symptoms, such as headache, eyestrain, and difficulty focusing, should be included in the definition of simulation sickness. This category of symptoms, sometimes called asthenopia, is not common in other forms of motion sickness. Anomalies in the simulator visual system are believed to be responsible for asthenopia. One argument is to include this cluster of symptoms, in which case the term "simulator induced syndrome" might be more appropriate than "simulator sickness." The other argument is to exclude asthenopia and reserve the term "simulator sickness" strictly for the motion sickness-like symptoms that sometimes result from exposure to simulation. Further discussion is needed to resolve this issue.

Flashbacks, characterized by apparent motion or rotation of the visual field, have been reported infrequently, sometimes many hours after the simulator exposure. [1] This type of symptom is rare and not widely documented, but remains a concern due to the presumed relevance for the safety and health of the affected aviator.

Brief History

Reports of motion sickness-like symptoms in ground-based flight trainers were flight documented in the U.S. Navy's 2-FH-2 Hovering and Autorotation Trainer more than 30 years ago. [2] Little research activity was directed toward this issue in the next 20 years. Since the late 1970's, many attempts have been made to determine the extent of the problem, identify the causal factors, and provide guidelines for its alleviation. [3,4,5] More recently, experimental efforts have attempted to identify characteristics of the simulator, the simulated flight scenario, and the simulator user which may lead to the occurrence of simulator sickness. [6, 7, 8] The issues involved in understanding and alleviating simulator sickness are relevant to the design and use of other systems which rely on the use of virtual imagery to represent orientation and motion through space, such as helmet mounted displays, advanced cockpits and crew stations, and remotely piloted vehicles.

Problem Identification

Several areas of concern were identified by the committee as being negatively influenced by the occurrence of simulator sickness:

(1) Safety and Health. There are potentially profound safety and health implications for users who experience either prolonged or delayed-onset symptoms following use of the simulator. Although members of the committee agreed that reports of long-term aftereffects of simulation are generally anecdotal and uncommon, the obvious implications for safety of flight require that the phenomena be assessed more thoroughly. The concern is that simulator sickness, or adaptation to the circumstances that produced the sickness, may increase the likelihood of a subsequent flight accident. A related concern is whether the simulator user is at risk while driving an automobile or engaged in similar tasks. Simulator manufacturers and simulation facility managers may be concerned about liability issues related to the frequency, nature, strength, and duration of simulator aftereffects. Very little information is available on prolonged or delayed-onset symptoms or long term simulator aftereffects.[1]

(2) Training Effectiveness. Negative training could result from simulator sickness. Strategies used by individuals to minimize the occurrence of sickness in the simulator (e.g., limiting head movements) could result in poor transfer of training to the aircraft. Also, user acceptance of a training simulator may suffer when a high incidence of sickness is associated with its use.

(3) Validity of R&D Data. Research and engineering design data could be contaminated if simulator pilots are experiencing adverse symptoms. However, committee members familiar with the use of simulators as engineering design tools report infrequent problems with simulator sickness. Reports of sickness in research simulators (in studies unrelated to simulator sickness) have been rare.

(4) Scheduling and Utilization. Establishing schedules for training flights and simulator sessions is complicated by simulator sickness. Pilots who suffer from severe symptoms may need to be removed from flight duties temporarily. Their sudden unavailability detracts from flight training schedules and, perhaps, on general flight readiness. Temporary grounding policies have been adopted as a precautionary measure following simulated flights at some military training locations. NAS Miramar, for example, has adopted a mandatory 12-hour grounding period following an initial training session in the F-14 simulator, and a 2-hour minimum following all subsequent simulator training sessions. The U.S. Army has,

in most cases adopted a 6-hr hour "waiting period" between simulator usage and actual flight.

The concerns listed and discussed above are speculative. They provide a rationale for further investigation of the simulator sickness phenomenon, which the Steering Committee views as an unwanted side effect of simulation. The Steering Committee members are unanimously strong advocates of simulation for training and research.

Incidence and Symptomatology

Although reports of simulator sickness have been documented with increasing frequency in recent years, some disagreement remains about the operational significance of the problem. Some members of the Steering Committee, for example, felt that strong evidence for simulator sickness requires comparison to the incidence of sickness occurring in the actual aircraft.

Committee members from the U.S. Navy and Army, as well as the Royal Air Force of the United Kingdom, described their efforts to quantify the occurrence of simulator sickness in their training facilities. In general, their approach has been to conduct on-site evaluations to assess the well-being of pilots before and after using the simulator in representative scenarios. These assessments generally have relied upon the use of some variant of the Motion Sickness Questionnaire (MSQ) developed more than 20 years ago.[9] The MSQ is essentially a checklist of 20 to 30 "major" and "minor" symptoms of conventional motion sickness. A single score is derived from the specific symptoms selected and their severity. Tests of postural stability also are typically used in assessing simulator sickness.[10]

The use of the MSQ and tests of postural equilibrium are employed in a pre-test, post-test fashion to assess changes in pilots' well-being following simulator training sessions. Representatives of the U.S. Navy stated that the use of pre- and post-measures may contribute to an underestimate of the incidence because some pilots may experience symptoms in the simulator and subsequently, through adaptation, recover before the end of the simulated flight.

The U.S. Navy has conducted simulator site surveys for over five years, and has generated a database of over 2000 observations at 10 simulator sites.[11] Their results reveal incidence rates (based on the existence of at least one symptom) ranging from 10% in the F-14 Weapon System Trainer at NAS Miramar to 60% in the SH-3 Operational Field Trainer at NAS Jacksonville. The U.S. Army has recently begun to assess the extent of the simulator sickness problem in their trainers, and has reported a 44% incidence rate in the AH-64 trainer at Ft. Rucker.[12] As a result of their investigations, the Navy has recently produced a set of guidelines in the form of a "field manual" for simulator instructors and pilots to attempt to minimize occurrences of the problem.

Steering Committee members representing the U.S. Air Force reported that they have had few significant indications of simulator sickness problems, with the possible exception of the Simulator for Air-to-Air Combat (SAAC) at Luke AFB. It was reported that an estimated 20%-30% of pilots using the SAAC voluntarily report some type of adverse symptom, although no official statistics are compiled. In approximately 5% of exposures, the effects are sufficient to warrant temporary interruption of the training session. There is no policy of temporary grounding at Luke following use of the SAAC. In general, the Air Force reports no significant problem or concern with simulator sickness at their simulator facilities, although no systematic analysis of the issue has been published.

Several members of the Steering Committee suggested that the incidence of simulator sickness reported by the Army and Navy may be overestimated because a "case" is claimed even when only one minor symptom, such as sweating or fatigue, is reported. Whether such minor symptoms are valid indicators of simulator sickness was questioned. No consensus was reached by the Steering Committee as to the number or type of symptoms required to indicate a case of simulator sickness. Likewise, there was no general agreement on the severity of symptoms required to prompt concern for flight safety or operational readiness.

Although sweating and fatigue are common in motion sickness, they may occur for other reasons. Pilots routinely sweat while engaged in intense simulation sessions like air-to-air combat or night carrier landings, for example, although the sweating only occasionally appears to be related to sickness. Furthermore, the occurrence of fatigue may be related to situational characteristics of the simulation, such as excessive physical or cognitive effort, and should be distinguished from the sleepiness ("sopite syndrome") that is sometimes characteristic of motion sickness.[13]

The definition of symptom prevalence used to indicate an overall incidence of simulator sickness at a specific simulator site is still an issue. The practice of using one "minor" symptom to identify a case of sickness was viewed by some Steering Committee members as producing an overestimate of the simulator sickness problem at Army and Navy facilities. On the other hand, the Air Force may be underestimating the problem because pilots may be unwilling to voluntarily report adverse symptoms.

Comparisons Between Simulators and Aircraft

The Steering Committee strongly recommended that future analyses of simulator sickness incidence include comparisons of the symptoms experienced in actual and simulated flight. Simulator surveys to date have not compared simulator and aircraft data for either of the common measures of simulator sickness, self-report (MSQ symptom ratings) or postural stability measurement.

Using air sickness data as a baseline will enable a more accurate assessment of the extent of the simulator sickness problem. This concept is inherent in the common definition of simulator sickness as a constellation of signs and symptoms that occurs in the simulator but not in actual flight. The Army and Navy have recently modified their simulator sickness questionnaires to determine whether maneuvers performed in the simulator result in adverse symptoms that would normally not be expected in the aircraft.

Conflict Theory

The commonly accepted theory of motion sickness is the sensory conflict theory, sometimes known as "neural conflict" or "neural mismatch." While these terms usually are considered to be synonymous, some Steering Committee members make a distinction between them. For example, it was suggested that the terms "sensory" and "neural" imply different levels of human processing of spatial information. Both are based on a temporal and/or spatial mismatch of information about one's orientation or motion through space, but sensory conflict implies that the mismatch occurs at the level of the proprioceptive end organs, while neural conflict implies a discrepancy between actual and expected information. Sensory conflict occurs at the receptors that directly receive information about orientation from the environment (primarily the visual and vestibular receptors). Neural conflict refers to a mismatch between the currently experienced pattern of proprioceptive stimulation, and a neural store of previously experienced patterns.

Both versions of the conflict theory have some strengths, but share a common weakness -- the lack of a quantifiable measure of conflict. Steering committee member John B. Sinacori suggested an initial approach to quantifying the physical conflict between the motion implied by the simulator visual system and motion delivered by the motion base. Briefly stated, a measure of conflict present in a simulated flight scenario could be defined as the integrated absolute value of the difference between the angular velocity and specific force vectors as calculated by the host computer and as produced by the visual and motion-base systems. In other words, the physical conflict would be calculated as the difference between the intended aircraft motion (portrayed by the visual system) and the actual motion produced by the motion base. Mr. Sinacori suggested this approach as a first attempt at the quantification of conflict theory, a process strongly recommended by the Steering Committee.

Consensus Items

The following list of consensus items is derived from highlights of the presentations given at the meeting as summarized by Dr. Fred E. Guedry, Jr.:

(1) There is a simulator sickness problem. Nausea, however, is not perceived as being as critical a problem as potential long term aftereffects.

(2) Several implications of simulator sickness or simulator aftereffects are important:

- Increased probability of in-flight accidents.
- Negative transfer of training.
- Increased risk of post exposure personal injury.
- Decreased user acceptance of training simulators.
- Compromised validity of data from research simulators.

(3) Temporal effects of perceptual adaptation are important. There is a need for more information about the rate of adaptation to the simulator and the rate of readaptation to the aircraft.

(4) The physical features that provoke sickness need to be clearly identified.

(5) An experimental approach is necessary to identify contributing factors.

(6) Quantification of the conflict theory is needed.

(7) Prediction of individual differences in susceptibility is recommended.

Directions for Future Work

In addition to the consensus items identified above, several suggestions were contributed to provide direction for future investigation of the simulator sickness problem. These are summarized briefly as follows:

(1) In conducting assessments of the incidence of simulator sickness at a particular facility, or when conducting research on simulator sickness, it is important to provide a sufficient engineering analysis of the simulator being used. This should include a fundamental technical description of the facility, including the motion base, visual displays, and temporal characteristics of both. These descriptions should be standardized to facilitate the pooling of data from different facilities.

(2) There is a need to develop a standardized vocabulary of technical terms relating to simulation and simulator sickness. The multidisciplinary interest in simulator sickness underscores the importance of generating a common language to facilitate accurate communication.

(3) A standardized means of assessing the presence of effects before, during, and

after simulator sessions is needed to support the formation of a global simulator sickness database. This includes the need to develop a reliable means to assess the nature and severity of long-term aftereffects.

(4) Any simulator sickness assessment or analysis needs to interpreted with respect to what happens in the actual aircraft performing similar maneuvers. By definition, simulator sickness does not exist if pilots become ill in the aircraft in the execution of similar scenarios. A comparison of incidence rates in the simulator and aircraft would provide a more definitive estimate of the extent of the simulator sickness problem.

(5) There is a need to develop more sensitive measures of simulator sickness, particularly with respect to postural equilibrium. The commonly used tests of standing and walking steadiness may be rather insensitive to ataxic side effects as compared to the level of precision available with a force platform or stabilometer. Similarly, the search for reliable physiological indices of sickness should be continued.

(6) Comprehensive descriptions of simulator scenarios are also necessary for adequate definition of the conditions encountered by the pilot. These descriptions, ideally, should include the actual and implied motion characteristics experienced by the simulator pilot during the scenario.

(7) Individual differences in susceptibility to simulator sickness may be predictable. Areas such as perceptual sensitivity, control strategies, personality and flight experience should be investigated.

(8) A database of engineering and procedural "fixes" that have been useful in alleviating simulator sickness should be developed. For instance, personnel operating the SAAC noted that they had more problems with sickness when there was a great deal of perceived roll in a scenario. After blurring the horizon on the visual display, they noted a subsequent decrease in the occurrence of sickness. These individual "lessons learned" need to be compiled in a single, widely-available database.

(9) Perceptual adaptation occurs with repeated exposures to the simulator, resulting in a reduction of simulator sickness. However, this may be achieved at the cost of readapting to the environment outside the simulator. For example, Gower et al observed a concurrent decrease in postural stability outside the simulator with reduction in symptomatology.[12] Better understanding is needed of the processes of perceptual adaptation related to simulator exposure followed by readaptation to the real world.

(10) The consensus of the committee was that the occurrence of transient adverse symptoms in the

simulator is less a matter of concern than the safety implications of long-term or delayed-onset aftereffects. The investigation of the nature, duration, and causes of aftereffects is a high-priority research issue.

(11) An experimental approach is recommended in addition to continued monitoring (survey) of the incidence of the problem at training simulation facilities. Empirical data are needed to identify the causal factors. Most military flight simulator facilities are devoted to training and are unavailable for experimental studies. Research simulation facilities, such as those at NASA Ames Research Center and the Navy's Visual Technology Research Simulator, represent ideal environments for gaining further understanding of the causal factors.

References

1. Ungs, T.J. (1988). Simulator induced syndrome in Coast Guard aviators. Aviation, Space, and Environmental Medicine, 59, (3), 267-272.

2. Havron, M.D. & Butler, L.F. (1957). Evaluation of training effectiveness of the 2-FH-2 helicopter flight trainer research tool (Technical Report NAVTRADEVCEN 1915-00-1). Port Washington, NY: Naval Training Device Center.

3. Casali, J.G. (1986). Vehicular simulation-induced sickness. Volume I: An overview (Technical Report NTSC-TR-86-010. Orlando, FL: Naval Training Systems Center.

4. Kennedy, R.S., Hettinger, L.J., & Lilienthal, M.G. (1989). Simulator sickness. In G.H. Crampton (Ed.), Motion and space sickness. Boca Raton, FL: CRC Press.

5. McCauley, M.E. (1984). Research issues in simulator sickness: Proceedings of a workshop. Washington, DC: National Academy Press.

6. Frank, L.H., Casali, J.G., & Wierwille, W.W. (1988). Modeling operator control performance and well-being as a function of simulator visual and motion system transport delays. In Motion cues in flight simulation and simulator induced sickness (AGARD Conference Proceedings No. 433). Neuilly Sur Seine, France.

7. Hettinger, L.J., Berbaum, K.S., Kennedy, R.S., Dunlap, W.P., Nolan, M.D., & Fowlkes, J.E. (Under preparation). Vection and simulator sickness.

8. Uliano, K.C., Kennedy, R.S., & Lambert, E.Y. (1986). Asynchronous visual delays and the development of simulator sickness. In Proceedings of the Human Factors Society 30th Annual Meeting (pp. 422-426). Dayton, OH: Human Factors Society.

9. Kellogg, R.S., Kennedy, R.S., & Graybiel, A. (1965). Motion sickness symptomatology of labyrinthine defective and normal subjects during zero maneuvers. Aerospace Medicine, 36, 315-318.

10. Graybiel, A. & Fregly, A.R. (1966). A new quantitative ataxia test battery. Acta Otolaryngologica, 61, 293-312.

11. Kennedy, R.S., Berbaum, K.S., Allgood, G.O., Lane, N.E., Lilienthal, M.G., & Baltzley, D.R. (1988). Etiological significance of equipment features and pilot history in simulator sickness. In Motion cues in flight simulation and simulator induced sickness (AGARD Conference Proceedings No. 433). Neuilly Sur Seine, France.

12. Gower, D.W., Lilienthal, M.G., Kennedy, R.S. & Fowlkes, J.E. (1988). Simulator sickness in U.S. Army and Navy fixed-and rotary-wing flight simulators. In Motion cues in flight simulation and simulator induced sickness (AGARD Conference Proceedings No. 433). Neuilly Sur Seine, France.

13. Graybiel, A. & Knepton, J. (1976). Sopite syndrome: A sometime sole manifestation of motion sickness. Aviation, Space, and Environmental Medicine, 47, 873-882.

89-3269-CP

SIMULATOR SICKNESS ON THE INCREASE

R. S. Kennedy
Essex Corporation
Orlando, FL

G. O. Allgood
Martin Marietta Energy Systems, Inc.
Oak Ridge, TN

M. G. Lilienthal
Naval Training Systems Center
Orlando, FL

Abstract

The usefulness of innovations in simulation technology may be compromised by a poorly understood phenomenon, viz., simulator sickness. Simulator sickness refers to motion sickness-like symptoms that occur in aircrew during and following training. This paper will: 1) describe and summarize the implications of simulator sickness, and 2) discuss a biocybernetic approach to the control of the problem.

Introduction

The usefulness of innovations in simulation technology may be compromised by a poorly understood phenomenon, viz., simulator sickness. Simulator sickness refers to motion sickness-like symptoms that occur in aircrew during and following training. Symptoms include general discomfort, stomach awareness, nausea, disorientation and fatigue. There is also a prominent component of visually-related disturbances such as eyestrain, headache, difficulty focusing and blurred vision. In this respect, simulator sickness resembles disturbances produced by wearing distorting, reversing, or inverting lenses. This implicates simulator visual display systems as contributory to the malady. Aftereffects associated with simulator sickness include postural instability, dizziness and flashbacks. Flashbacks, which include illusory sensations of climbing and turning, sensations of negative g, and perceived inversions of the visual field, are particularly problematic because of their sudden, unexpected onset and risk to safety (1, 2).

In addition to risk to pilot safety, other implications of simulator sickness include training and operational readiness. Simulator sickness may compromise training. Aircrew who develop negative opinions because of simulator sickness will use simulators less or accomplish fewer training objectives. Finally, enforced restrictions from flying following training in some simulators may limit pilot operational readiness. This paper will: 1) describe and summarize the implications of simulator sickness, and 2) discuss a biocybernetic approach to the control of the problem.

Increase of Simulator Sickness

Simulator sickness was first reported by Havron and Butler (3) in 1957 in their evaluation of the 2FH2 (Bell HTL-4 helicopter) training device. Since that time, it has been reported in devices simulating fixed and rotary wing aircraft (4,5,6,7), automobile simulators (8,9,10), and tank simulators (11). It is pervasive in that it occurs in both fixed- and motion-base devices, and in devices with a variety of projection and image generation techniques. In general, virtual image, multiple cathode ray tube (CRT) systems characterize systems associated with the highest incidence of simulator sickness (6) which is one reason while the Navy chose a dome-based visual system for the V-22 simulator under development. The problem of simulator sickness is likely to be aggravated with advances in technology and trends for representing simulated flight in a more realistic fashion. With advances in computer image generation technology, near photographic quality of virtual worlds can be created. However, such developments must occur in tandem with research on the effects of such technology on the users.

Contributory Factors

The occurrence of simulator sickness is a "flag" that something is wrong with the system (Figure 1). Experts in the field are convinced that the answer to simulator sickness will not be in the form of one solution. Engineering and simulator usage factors implicated in simulator sickness include:

Engineering Factors

o Calibration of the Visual System

Improper calibration of virtual image display systems may lead to excessive accommodative and vergence demands, unequal accommodative demands between the two eyes, as well as conflicting vergence and accommodative responses, all of which may produce symptoms of eyestrain (12). Excessive flicker, misalignment of adjacent virtual image channel displays, and dynamic visual distortions may also produce asthenopic symptoms.

o Calibration of the Motion Base

Navy research has examined the effects on sickness rates of differing energy spectra in moving base simulators (13). The results showed that incidence of sickness was greater in a simulator with energy spectra in the region described as nauseogenic by the 1981 Military Standard 1472C (14). The high sickness rates were possibly experienced as a function of time exceeding these very low frequency limits, although pilot and/or equipment factors not related to the motion base may provide alternative explanations or be contributory factors.

o Visual/Motion Coordination (Synchronization and Lag)

Computational limitations generally produce temporal lags between operator control input and subsequent changes in position as indicated by the visual display and motion base. Whether such

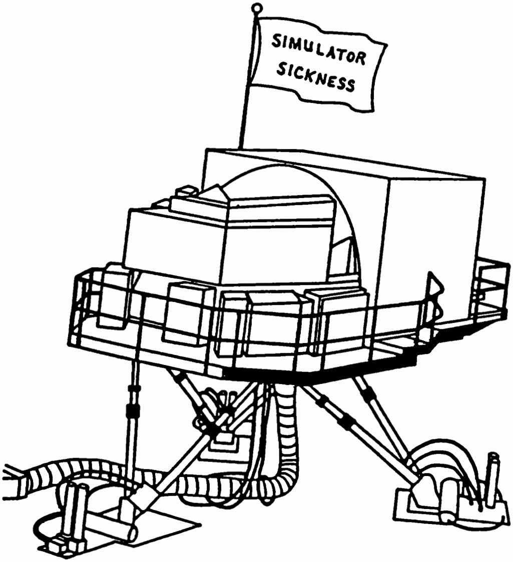


Figure 1. Simulator Sickness may Serve as a "Flag" that Something is Wrong with the Simulator

delays are conducive to simulator sickness has not been extensively studied (15). However, it is known that lags may cause pilot induced oscillations which can have two consequences: 1) overtax the visual system and create dynamic visual distortions, and 2) produce nauseogenic inertial energy around .2 Hz.

Simulator Usage Factors

- Flight Scenarios

Excessive close ground interaction, particularly when turning or taxiing, and rapid changes in altitude may be conducive of simulator sickness. Modification of flight scenarios may be one of the most effective methods for controlling simulator sickness (16). However, this method of controlling the problem may be unacceptable in that the simulator may not be used to achieve training objectives.

- Freeze

Improper use of the freeze and reset functions are associated with simulator sickness (16) Freezes should be effected only after straight and level flight has been achieved or after flying into the clouds.

- Hop Length

Long exposure periods may be conducive of simulator sickness (16). Crosby and Kennedy (17) showed that when aircrew members took breaks midway through a four hour simulator flight scenario, far fewer symptoms were experienced than when they had the full four hour session without a break.

Biocybernetic Approach: On-Site Monitoring

An approach to the control of simulator sickness is on-site quality control tracking of simulators such as is outlined in Table 1. A monitoring system can be installed at a simulator facility to collect a cumulative record of pilot reactions (e.g., symptomatology). A baseline can be set which, if exceeded, would serve as a flag or signal that the system needs to be evaluated. Subsequent evaluation would include collecting engineering data (e.g., the presence of very low frequency nauseogenic inertial energy, optical characteristics of visual display channels), and detailed human symptomatology data. The integration of such data would lead to recommendations or fixes. Such recommendations could be incorporated into simulator design specifications for future simulation devices.

We currently have a prototype of such a system implemented at NAS Whiting Field, FL, for the TH-57C (device 2B42) trainer. Following simulator training, aircrew take a computer implemented questionnaire which is presented on a portable Zenith microcomputer (Figure 2). The questionnaire takes approximately three minutes to complete: Aircrew first enter background data (e.g., state of fitness) and then rate the degree of severity of 16 symptoms. The symptomatology data are automatically scored (18) and if the score exceeds a criterion value, the student pilot is told to see his flight instructor who makes sure that the pilot remains on-site until the symptoms subside. Data obtained with the device may be sorted by student pilot, pilot function (i.e., pilot or copilot), and by the simulator device in which training was conducted.

In addition, a running average may be calculated across simulator flights to obtain data regarding increases or decreases in sickness that occur. Figure 3 shows a running average of data collected by the device over a time period of October through December 1988. Each point in the figure represents the average across 10 days of use of the simulator, such that the first point represents the average of days 1 through 10, the second point days 2 through 11, etc. The data show that sickness in the device systematically decreased after training with the device was initiated, indicating perhaps that the users were regulating use of the system (through control of scenarios) to reduce symptomatology. However, it is also apparent that after some period of time, sickness increased, which may represent the introduction of new syllabus components or indicate that maintenance is required. It is such increases in sickness that may be used to alert system users that the system needs to be evaluated and that aircrew may be at risk. Such a mechanism will be implemented in the next version of the device.

Conclusions

Higher fidelity simulation is needed in order to train more tasks more effectively. However,

Table 1. Detection, Diagnosis, Prescription of Simulator Sickness Problems

1) IS THE SYSTEM SICK

- On-site "Quality Control" Tracking of Overall Pilot Reactions

2) WHAT'S WRONG WITH IT (DIAGNOSIS)

- Engineering Data
- Symptomatology Data
- Other Human Data (e.g., Aftereffects)

3) HOW DO WE FIX IT? (PRESCRIPTION)

- "Expert System" Routines
- Integrate Pilot Symptoms, Pilot History, Engineering Data by Rules
- Make Recommendations for Fixes
- Feedback Recommendation to Simulator Design Specifications

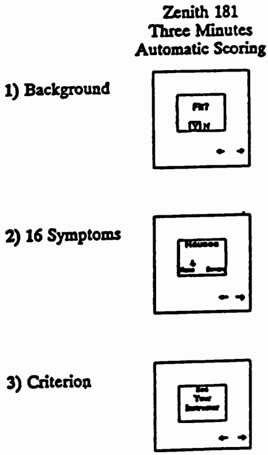


Figure 2. Description of the Reporting Device

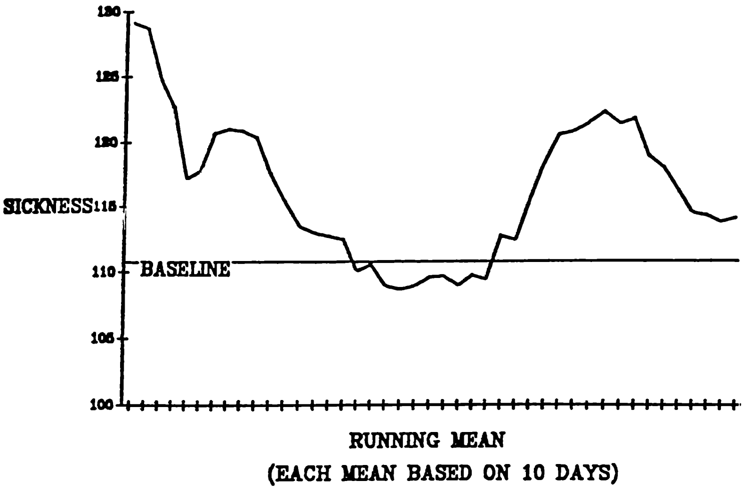


Figure 3. Running Average of Symptomatology over Simulator Exposures
(100 indicates absence of symptomatology; 112 indicates maximum level of acceptable sickness)

technological developments cannot be uncritically applied. In concert with such developments, human engineering advances must also be made so that untoward effects of engineering technologies on humans can be avoided. Biocybernetic monitoring of simulator sickness may serve as a interim approach to the control and understanding of simulator sickness.

References

1. Baltzley, D. R., Kennedy, R. S., Berbaum, K. S., Lilienthal, M. G., & Gower, D. W. (1989, in press). The time course of post-flight simulator sickness symptoms. Aviation, Space, and Environmental Medicine.

2. Kellogg, R. S., Castore, C. H., & Coward, R. E. (1980). Psychophysiological effects of training in a full vision simulator. Proceedings of the 51st Annual Meeting of the Aerospace Medical Association, Anaheim, CA.

3. Havron, M. D., & Butler, L. F. (1957). Evaluation of training effectiveness of the 2FH2 helicopter flight training research tool (Technical Report No. NAVTRADEVCEN 1915-00-1). Port Washington, NY: Naval Training Device Center.

4. Crowley, J. S. (1987). Simulator sickness: A problem for Army Aviation. Aviation, Space, and Environmental Medicine, 58, 355-357.

5. Gower, D. W., Lilienthal, M. G., Kennedy, R. S., Fowlkes, J. E., & Baltzley, D. R. (1987). Simulator sickness in the AH-64 Apache Combat Mission Simulator (USAARL Report No. 88-1). Fort Rucker, AL: United States Army Aeromedical Research Laboratory.

6. Kennedy, R. S., Lilienthal. M. G., Berbaum, K. S., Baltzley, D. R., & McCauley, M. E. (1989). Simulator sickness in U. S. Navy Flight Simulators. Aviation, Space, and Environmental Medicine, 60, 10-6.

7. Ungs, T. J. (1988). Simulator induced syndrome in Coast Guard Aviators. Aviation, Space, and Environmental Medicine, 59, 267-272.

8. Barrett, G. V., & Thornton, C. L. (1968). Relationship between perceptual style and driver reaction to an emergency situation. Journal of Applied Psychology, 52, 169-173.

9. Reason, J. T., & Diaz, E. (1971). Simulator sickness in passive observers (FPRC/1310). London, England: Ministry of Defence, Flying Personnel Research Committee.

10. Casali, J. G., & Wierwille, W. W. (1980). The effects of various design alternatives on moving-base driving simulator discomfort. Human Factors, 22, 741-756.

11. M. G. Lilienthal, Personal communication

12. Ebenholtz, S. E. (1988). Sources of asthenopia in Navy flight simulator (Report No. 87-635). Orlando, FL: Naval Training Systems Center.

13. Van Hoy, B. W., Allgood, G. O., Lilienthal, M. G., Kennedy, R. S., & Hooper, J. M. (1987). Inertial and control systems measurements of two motion-base flight simulators for evaluation of the incidence of simulator sickness. Proceedings of the IMAGE Conference IV. Phoenix, AZ: Image Society Incorporated.

14. Military Standard 1472C, Human factors engineering design criteria for military systems and facilities (MIL-STD-1472C). Washington, D.C.: Department of Defense, 1981.

15. Uliano, K. C., Lambert, E. Y., Kennedy, R. S., & Sheppard, D. J. (1986). The effects of asynchronous visual delays on simulator flight performance and the development of simulator sickness symptomatology (Preliminary Final report NAVTRASYSCEN 85-D-0026-1). Orlando, FL: Naval Training Systems Center.

16. Kennedy, R. S., Berbaum, K. S., Lilienthal, M. G., Mulligan, B. E., & Funaro, J. (1987). Guidelines for alleviation of simulator sickness symptomatology (NTSC-TR-87-007). Orlando, FL: Naval Training Systems Center.

17. Crosby, T. N., & Kennedy, R. S. (1982). Postural disequilibrium and simulator sickness following flights in the P3-C operational flight trainer. Proceedings of the 53rd Annual Aerospace Medical Association, Bal Harbor, FL.

18. Lane, N. E., & Kennedy, R. S. (1988). A new method for quantifying simulator sickness: Development and application of the simulator sickness questionnaire (SSQ) (EOTR 88-7). Orlando, FL: Essex Corporation.

89-3271-CP

MODEL-BASED GUIDELINES FOR SIMULATOR TEMPORAL FIDELITY REQUIREMENTS

William H. Levison*
BBN Systems and Technologies Corporation
Cambridge, Massachusetts

Abstract

Research was conducted to provide model-based entries for a forthcoming handbook on simulator temporal fidelity. Model analysis of attitude step-tracking tasks showed that the sensitivity of the performance/delay relationship to the task-related factors of vehicle response and performance requirements could be greatly reduced through appropriate normalization of both tracking performance and simulator delay. A linear rule-of-thumb for predicting effects of delay was derived in which performance is normalized with respect to the performance predicted for the baseline (no delay) task, and delay is normalized with respect to the reciprocal of the gain-crossover frequency for the baseline task.

Nomenclature

CWI	Control workload index
ISE	Integral-squared error
OCM	Optimal control model for pilot/vehicle systems

Introduction

Because of the extensive use of flight simulators as both training and engineering evaluation devices, considerable effort has been devoted to quantifying differences that exist between the simulation and in-flight environment and to minimizing the impact of these differences on the intended results of the simulator activity. Temporal distortion in the response of the simulator to pilot control inputs -- more commonly referred to simply as "simulator delay" -- has received perhaps the greatest attention as a source of simulator infidelity. Much of the research on simulator delay has been aimed toward the development of guidelines on how much simulator delay can be tolerated in a given simulation environment.

The work reported in this paper was motivated specifically by a desire to generate a "rule of thumb" for assessing the significance of simulator delay and, more generally, to provide model-based entries for a handbook that is expected to contain a comprehensive summary of information relating to temporal fidelity in flight simulators.**

The formulation of rules of thumb and other guidelines for simulator delay is complicated by the dependency of the performance and workload consequences of delay on a variety of procedural and task-related factors. Procedural factors include (1) the definition of "delay", (2) the intended use of the simulator, (3) the performance dimension(s) along which the effects of delay are to be measured or predicted, and (4) the range of delay-related performance decrement that is considered "acceptable". For the purposes of this paper, we delimit these factors as follows:

- Simulator "delay" is associated with the phase lag appearing in the simulator response that is in excess of the phase lag that would be present in the operational situation. If this phase shift is linear with frequency(as is the case for the tasks analyzed in this paper), the "delay" is identical to the transport lag that would exhibit this phase characteristic. If this phase shift is caused by low-pass filtering alone or in combination with pure transport delay, procedures such as those suggested by Smith and Bailey[1] will be needed to define the appropriate "delay".
- The details of the simulation experiment or model analysis depends on whether the simulator is to be used as an engineering evaluation tool (e.g., flying qualities assessment) or as a training device. Since the modeling effort reported in this paper was directed toward predicting performance in situations where the pilot was assumed to be well-trained, the results presented later are more oriented toward the simulator as an evaluation tool. If the simulator is to be used as a training device, comparison will typically be made between a situation in which the pilot is well trained (simulator with delay) and a situation in which the pilot may not be well trained (first few exposures to the operational system).
- Performance dimensions considered in this paper are mean-squared tracking error scores and pilot opinion ratings for the well-trained pilot. If the focus were on training simulators and transfer-of-training issues, it would be more appropriate to explore performance as a function of exposure to the task, especially after the pilot has transitioned from the

* Divisional Scientist, Senior Member AIAA

**"Simulator Temporal Fidelity: A Guide for Research and Development", Armstrong Aerospace Medical Research Laboratory, (in preparation) training environment to the operational environment. Other measures such as pilot frequency response and subjective workload would be relevant to both types of simulator application.

- The criterion for "acceptable" delay is deferred. It is left to the reader to decide how much in terms of delay-related changes in objective or subjective performance can be tolerated. Typically, this decision will be based on a tradeoff analysis involving factors, such as cost, that are beyond the scope of this paper.

In addition to these procedural factors, a number of task-related factors will determine the significance of simulator delay in a particular situation. Numerous experimental and analytical studies have explored the interaction between vehicle response characteristics and delay effects with inconsistent results[2-9]; some studies have found greater delay effects for fast-responding vehicles, whereas others have found greater effects with sluggish vehicles. These results may have been confounded by the apparently substantial influence of task requirements on performance degradation due to delay, with deleterious effects increasing with tighter control requirements[2,3,5,9,10].

Other factors potentially influencing the effects of delay include the location of the delay within the pilot-vehicle system (e.g., a control-system delay that appears in the innermost control loop versus a computational delay appearing only in the presentation of an outer loop variable such as path error); the qualitative nature of the external input (transient versus steady-state or random input, command versus vehicle disturbance); the quantitative nature of the external input (amplitude and spectral content); and the qualitative nature of the primary flight task (attitude versus flight-path regulation). Delay effects have been found experimentally and analytically to be different for fixed- and moving-base simulations[3,5]. One study has found that the effects of an equivalent delay associated with a force-feel system were substantially less than the effects of the same equivalent delay introduced electronically in the control loop[11].

Yet another issue -- one not considered in this paper -- is that of cue asynchrony in which cues provided by the visual and platform motion systems are delayed differently. In the following discussion we assume that all cues affected by simulator delay are delayed by the same amount.

The development of guidelines for simulator delay, whether based on experimental data or model predictions, will be made more tractable if the dimensionality of the problem can somehow be reduced. To this end we offer model-based evidence that the sensitivity of the performance/delay relationship to certain task factors can be appreciably reduced if both performance and simulator delay are suitably normalized. Specifically, we normalize performance predictions in the presence of delay to performance that is predicted in the absence of delay, and simulator delay is normalized with respect to a characteristic response time of the closed-loop system. Normalization of performance is based on the assumption that delay acceptance criteria will ultimately be based on the percentage difference in performance and/or workload between the simulator with delay and the operational system. The rationale for normalizing simulator delay is that it "works"; i.e., it appears to compress the data.

The idea of normalizing performance and delay is not new; the approach suggested in this paper is encouraged in part by the apparent success of Allen and DiMarco[2] and Sinacori[12] in normalizing their results. To the knowledge of the author, however, this paper is the first to provide evidence of data compression -- either experimental or model-based -- in which normalized tracking error scores are plotted versus normalized simulator delay.

Definitions and Concepts

The following frequency-domain metrics and concepts commonly used in describing the performance of linear closed-loop control systems are referenced in this paper:

- Open-loop describing function (dimensionless): the combined operator/vehicle frequency response, which in this paper is equivalent to the control-return difference (i.e., the effective transfer from a control input to a resulting commanded control correction).

- Gain-crossover frequency (rad/sec): the lowest frequency at which the magnitude of the open-loop describing function is unity.

- Phase margin (degrees): 180 degrees minus the phase lag of the open-loop describing function at the gain-crossover frequency. Phase margin is positive for stable closed-loop systems.

- Gain margin (dimensionless): 1.0 minus the gain of the open-loop describing function at the frequency at which the phase shift of the describing function is -180 degrees, usually expressed in dB.

The following additional concepts are defined here:

- Delay Margin (seconds): the amount of additional open-loop delay required to drive the closed-loop system unstable in the absence of any phase or gain compensation. The delay margin equals the phase margin divided by the gain-crossover frequency.

- Stability margin (dimensionless): the closest distance of the open-loop describing function to the instability point (-1.0, 0.0). This concept has meaning only for stable closed-loop control systems.

- Stability-margin frequency (rad/sec): the frequency at which the open-loop describing function is closest to (-1.0, 0.0).
- Control Workload Index, or CWI (dimensionless): the ratio OL/SM, where OL is the magnitude of the open-loop describing function at the stability-margin frequency, and SM is the stability margin. OL is the actual magnitude; i.e., not converted to dB. For many control systems, the stability margin is approximately equal to the gain margin, and the CWI is approximately the open-loop gain divided by the gain margin.

The concept of control workload is introduced to provide a mechanism by which we can explore, through model analyses, the impact of piloting strategy (in terms of relative aggressiveness) on the relationship between closed-loop performance and simulator delay. Increasing the CWI implies reducing the stability and phase margins, "tightening" the control loop, and responding in a more aggressive manner. For the case in which the CWI is closely approximated as the open-loop gain divided by the gain margin, the CWI is the reciprocal of the fractional increase in control gain that can be tolerated before the system is driven unstable. The CWI is thus directly related to the required precision of control.

The following definitions are specific to the material that follows:

- Baseline task: the tracking task (vehicle plus pilot response) in the absence of experimental (simulator) delay added to system response. The baseline task may be defined to include irreducible aircraft or simulator delay. The word "baseline" used in conjunction with a system response variable (e.g., "baseline crossover frequency") refers to the value of that variable in the baseline task.
- Added delay: delay associated with pilot/vehicle response that is in excess of the delay associated with the baseline task. "Added delay" is distinguished from "simulator delay" in that a simulator task used as the baseline task may include irreducible delay components, whereas added delay is the component of delay that is considered to be the independent variable of the analysis or experiment.
- Relative error: predicted tracking error score normalized with respect to the error score predicted for the baseline task. Integral-squared error (ISE)is used as the performance metric for the tasks explored later in this paper.
- Margin-relative delay: added delay, normalized with respect to the delay margin associated with the baseline task.
- Crossover-relative delay: added delay, normalized with respect to the reciprocal of the gain-crossover frequency associated with the baseline task; equivalently, the added delay times the baseline crossover frequency.

Model Predictions of Time Delay Effects

Model analysis reported herein was performed with the optimal control model (OCM) for pilot/vehicle systems originally developed by Kleinman, Baron, and Levison[13,14]. This model form was chosen in part because it is readily formulated to yield predictions of a variety of closed-looped metrics (including mean-squared error scores and frequency-domain variables), and it has been found to replicate the effects of time delay[5,15,16] and other forms of simulator infidelity[17] on measured performance.

In the following analysis we explore the effects of added delay on predicted performance for attitude tracking tasks similar to those employed by Bailey et al. in their study of the effects of control-system delay on control performance and pilot opinion ratings[3]. These particular illustrative tasks are selected because (1) they explore multiple task parameters, (2) the experiments are well-documented, (3) the task -- attitude tracking -- is one of the standard tasks used experimentally to define aircraft flying qualities, and (4) the OCM has been shown to replicate the trends of the tracking error scores[5].

Description of the Tracking Task

In addition to simulator delay, principal experiment variables of the Bailey study were simulated aircraft type, simulator mode, and the details of the external forcing function. Simulated aircraft types ranged from dynamics representative of a high-performance fighter flown in a relatively aggressive manner (i.e., tight loop closure) to dynamics of a heavy transport flown in a relatively more benign task environment. Short-period characteristics of the aircraft types explored in the following analysis are shown in Table 1. The pilot's task was to minimize attitude tracking errors in response to command or disturbance inputs in the pitch and/or roll axes. Data were obtained in-flight using the NT-33A variable stability airplane and on the ground with the NT-33A in a fixed-base mode.

Table 1. Summary of Configuration Characteristics (from Bailey et al.)

Aircraft Type	Fighter	Transport
Short-Period Damping	0.70	0.40
Short-Period Natural Frequency (rad/sec)	6.30	2.0
Longitudinal Stick Gain (lbs/g)	10.0	30.0
Roll Mode Time Constant (sec)	0.35	1.0
Lateral Stick Sensitivity (lbs/in)	3.5	6.0
Irreducible Simulator Delay (Seconds)	0.09	0.09

Each evaluation flight contained segments in which the input consisted of a linked series of step and ramp commands deemed particularly suitable for handling qualities evaluations, and segments in which the input was a sum-of-sines command or disturbance intended to facilitate frequency-domain analysis of pilot response. Through manipulation of both the amplitudes and spectral content of the external inputs, and of the control gains, the pilots were encouraged to perform the transport tasks in a less aggressive manner than the fighter tasks.

Objective closed-loop performance metrics were obtained from the sum-of-sines segments, whereas Cooper-Harper opinion ratings were obtained for the entire flight. Model analysis of the continuous tracking segments replicated the major experimental trends of the objective performance measures[5]. Bailey et al. concluded from the opinion data that somewhat larger simulator delays could be tolerated for the less responsive vehicles, or for less demanding tasks, than for the more responsive vehicle flown in a demanding task environment. In the case of the fighter aircraft, simulator delay had a more adverse effect in the ground-base than in the in-flight trials.

Although the study by Bailey et al. used continuous and linked step-ramp inputs, the following analysis is performed for the task of minimizing integral-squared error in response to either a pitch- or roll-axis step command. The step-tracking task is chosen for two reasons. First, some of results of that study suggest that pilot opinion was based on vehicle or closed-loop response time rather than tracking performance. Specifically, a more adverse rating was assigned to the simulated transport than to the simulated fighter in the absence of experimental delay, even though lower rms error scores were obtained for the transport. Pilot commentary indicated that the long response time of the transport influenced the ratings.

A step command-following task provides a more suitable environment for exploring closed-loop response time than a steady-state random-input tracking task. Not only can one compute an effective response time from the tracking error waveform, but the integral-squared error, normalized with respect to the squared step input, has units of time and can thus be interpreted as a response-time metric. For the case in which the closed-loop response to a step input may be approximated as the output of a first-order system (i.e., simple exponential transient), the ISE equals half the system time constant.

A second reason for exploring step responses is that many of the published results of flying qualities studies against which to test further the notions presented in this paper pertain to tasks where the pilot may be assumed to have performed an explicit or self-generated step tracking task (e.g., acquire a target in attitude, perform a side-step maneuver).

Model analysis was performed to yield predictions of integral-squared error as a function of added delay and control workload index (CWI) -- the latter being a metric relating to the "aggressiveness" of the piloting task. In order to use the steady-state implementation of the OCM to make these predictions, a steady-state approximation to the transient task was formulated[18]. A low-pass filter having a very low critical frequency (0.01 rad/sec) was simulated to approximate an integrated noise process (the continuous-input equivalent of the transient step input), and the resulting mean-squared error prediction was considered to be numerically equivalent to the ISE that would be predicted for the corresponding step-tracking task. Use of the steady-state model formulation was preferred over use of a "simulation" implementation of the OCM because the steady-state model is generally more computationally efficient, it has been more widely validated, and it is more readily available to the potential user community.

Model Parameterization

For single-variable control tasks with control and display gains adjusted so that threshold and saturation phenomena may be ignored, the OCM is usually considered to have the following independent "pilot-related" parameters: (1) operator time delay, (2) observation noise/signal ratio, (3) motor noise/signal ratio, and (4) a "cost weighting" assigned to control-rate variance in a quadratic performance index defined as the sum of error variance plus weighted control-rate variance. (Readers unfamiliar with the structure and parameterization of the OCM are directed to the literature.[13,14])

The quadratic performance index consisted of a unity "cost weighting" on attitude error variance and a non-zero weighting on control-rate variance. With the control weighting set to zero, the control-rate weighting was first adjusted to provide minimum mean-squared error, then increased to a point where the error was 5% above the minimum. Because the motor noise was essentially treated as a multiplicative rather than a fixed additive noise (i.e., its variance was iteratively adjusted to scale with predicted control-rate variance), minimum error was obtained with a non-zero control-rate weighting. Because this minimum was rather broad, yielding a large range of control scores for essentially minimum error, a small increase in error beyond the computed minimum was accepted to keep the control score on the low side (a reasonable piloting strategy).

The CWI resulting from this parameter adjustment scheme --obtained via frequency-response analysis -- was considered to reflect the maximum control workload at which the pilot would find it useful to operate for the particular task. In order to force the (mathematical) pilot to operate at a lower workload (i.e., fly less aggressively), a penalty on control variance was added to the quadratic performance index, and the cost weighting associated with this quantity was manipulated until the desired CWI was obtained.

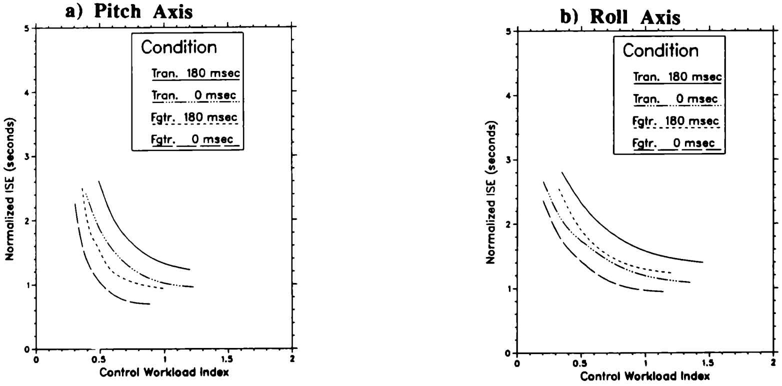
The perceptual variables assumed to be utilized by the pilot were selected to reflect the in-flight condition in which whole-body motion cues are present. Assumed perceptual variables for pitch tracking were pitch error, pitch attitude, pitch rate, pitch acceleration, and normal acceleration; perceptual variables for roll tracking were roll error, roll angle, roll rate, and roll acceleration. No attention-sharing penalties among perceptual variables were represented[19]. Other pilot-related model parameters were:

Operator delay = 0.2 sec.
Observation noise/signal ratio = -17 dB.
Motor noise/signal ratio = -38 dB

Model Predictions

Pitch and roll tracking task were analyzed separately. Figure 1 compares performance/workload tradeoffs for the two simulated vehicles for zero and 180 msec added delay, where "performance" is defined as predicted ISE, normalized with respect to the square of the step command, and "workload" is in terms of the CWI. Predictions for the pitch- and roll-axis step tracking tasks are given in Figures 1a and 1b, respectively. The "added delay", which was implemented as a pure transport lag in the control path, was in addition to the irreducible simulator delay of about 90 msec. For purposes of the analysis presented in this section, this residual delay is treated as part of the vehicle response characteristics in the sense that the baseline condition includes this delay.

The model predictions agree qualitatively with the flying qualities results reported by Bailey et al. in that for a given workload, performance degrades as added delay is increased, and, for a given delay, performance is worse for the transport than for the fighter. Conversely, to maintain a given level of performance, the required control workload increases with delay and is greater for the transport than for the fighter. These model predictions suggest that for tasks where the pilot attempts to operate at a maximally useful CWI in the baseline condition, the addition of transport delay will necessarily degrade tracking performance. One would expect in this case a correlation between objective performance metrics and pilot opinion rating. On the other hand, if the baseline task is "benign" in the sense that the pilot can operate well below maximum CWI, the pilot may accommodate additional delay by increasing the control workload so as to maintain the baseline performance level. In this case, pilot opinion rating may begin to degrade


a) Pitch Axis

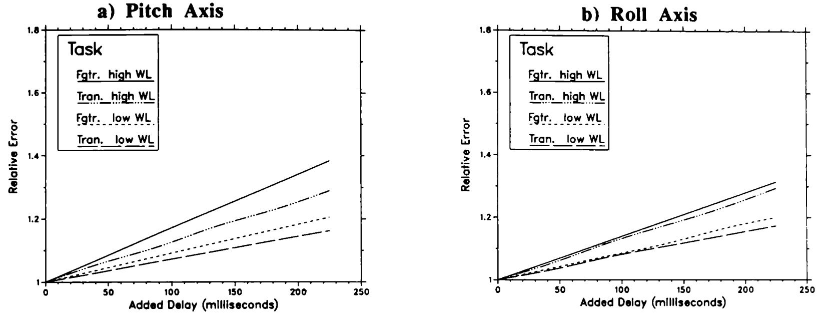
b) Roll Axis

Figure 1. Predicted Performance Workload Tradeoffs

before an effect is seen in terms of tracking error or closed-loop system response time.

If one were to base "acceptance" of simulator delay solely on absolute, objective performance levels, one would conclude from Figure 1 that, for the task conditions explored here, a given amount of added delay would be less tolerable for the transport than for the fighter, and less tolerable for low workload than for high workload conditions. As discussed earlier in this paper, however, we assume that delay acceptance will be based on one or more measures of performance -- be they objective or subjective -- relative to the measures that obtain in the appropriate baseline condition. Under this hypothesis, the impact of added delay on, say, an attitude tracking task with the transport flown in a relaxed (low CWI) manner should be based on performance in that same task environment with no added delay, not on performance using the same vehicle flown aggressively, or using some other vehicle.

Figures 2a and 2b shows the effects of added delay on "relative error", defined as the ISE score predicted for a given added delay normalized with respect to the score predicted for the same task with no added delay. Predictions are shown for the fighter and transport, each "flown" at two levels of CWI. The baseline "low workload" condition was simulated via adjustment of the control-variance cost weighting until the CWI was half of what is was for the baseline "high workload" condition with no penalty on control variance. Key frequency-response parameters for the conditions represented in Figure 2 are given in Table 2.


a) Pitch Axis

b) Roll Axis

Figure 2. Effects of Added Delay on Predicted Relative Error

Table 2. Key Frequency Response Parameters for the OCM Analysis

	Pitch				Roll			
	Fighter		Transport		Fighter		Transport	
	High WL	Low WL	High WL	Low WL	High WL	Low WL	High WL	Low WL
Crossover Freq	1.67	1.01	1.46	855	1.57	1.01	1.57	1.01
Delay Margin	485	1.01	542	1.12	450	820	377	715
Phase Margin	46	59	45	55	40	47	34	41
CWI	0.88	0.44	1.23	0.60	1.14	0.56	1.36	0.66

Predicted error relative to the corresponding baseline error increased in an approximate linear fashion with added delay for the range of delay analyzed (zero to 225 msec). Performance trends agree qualitatively with the conclusions drawn by Bailey et al. in that the adverse effects of delay are greater for the high-performance aircraft than for the more sluggish aircraft, and greater for a relatively aggressive strategy than for a relatively relaxed strategy. The effects of delay, stated as percentage increase in predicted error per 100 msec added delay, are shown in Table 3.

Table 3. Effects of Task Variables on Performance/Delay Relationship

Vehicle	Workload	% Increase/100 msec Pitch	Roll
Fighter	High	17	14
Fighter	Low	9	9
Transport	High	13	13
Transport	Low	7	8

Figure 2 and Table 3 show that assumed control workload had the largest influence on the performance/delay relationship, vehicle dynamics the next largest effect, and axis of control the least. The sensitivity of relative error to added delay varied by a factor of about 2.1 from the most sensitive condition (fighter, high workload, pitch control) to the least sensitive condition (transport, low workload, pitch control).

The results shown here are to some extent specific to the details of the assumed task conditions: different choices for CWI, simulated aircraft dynamics, tracking tasks, and perhaps pilot-related model parameters would yield different numbers and possibly different trends across the variables of the analysis. Nevertheless, it is clear that these factors can be expected to influence the sensitivity of performance (and presumably pilot opinion) to a given amount of simulator delay, and that some form of data compression is desirable if we are to derive general guidelines for acceptable delays.

Figure 3 shows that the data are compressed slightly when added delay is normalized with respect to the delay margin predicted for the baseline condition. Sensitivity varies by a factor of about 1.9 across the eight task conditions. Substantial further compression occurs when delay is normalized with respect to the inverse crossover frequency as shown in Figure 4. The sensitivity of relative performance to normalized delay now varies by a factor of about 1.4 across conditions -- a substantial compression compared to the variation in sensitivity found when added delay is not normalized.

The reduced task-dependency of the results when plotted against the crossover-relative delay suggests a rule-of-thumb for predicting the consequences of added delay on performance in a step tracking task. Taking the average of the maximum and minimum slopes shown in Figure 4 and rounding off to the nearest 5%, we see that the relative error increases by about 90% per unit of relative delay. Noting that "relative delay" as used in Figure 4 is simply the product of the added delay and the baseline gain-crossover frequency, we obtain the following rule:

$$\% \text{ increase in ISE} = 90 * \omega_c * \Delta T$$

where ω_c is the crossover frequency (rad/sec) of the closed-loop system in the absence of added delay, and ΔT is the added delay in seconds.

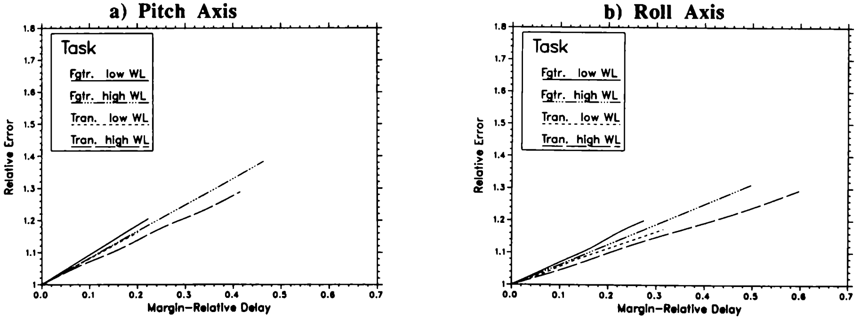
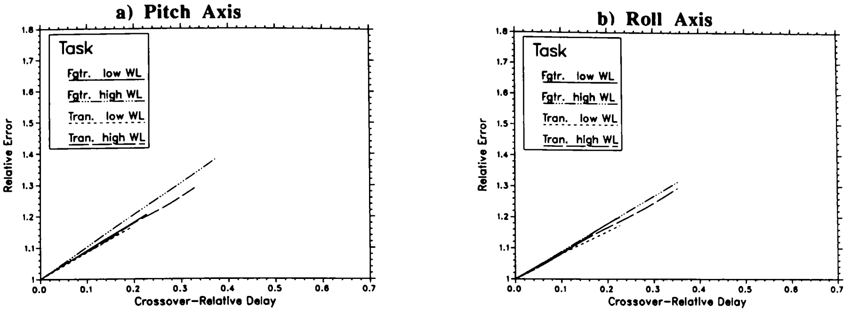


Figure 3. Effects of Margin-Relative Delay on Predicted Relative Error


a) Pitch Axis

b) Roll Axis

Figure 4. Effects of Crossover-Relative Delay on Predicted Relative Error

Discussion

For the particular task environments explored in this paper, normalization of both performance and simulator delay appears to have substantially desensitized the performance/delay relationship to the task-related factors of vehicle response dynamics and performance requirements (equivalently, response aggressiveness). The "price" paid for this data compression is that simulator delay is normalized with respect to a metric that is a function of combined pilot/vehicle response-- delay margin or gain-crossover frequency -- rather than a parameter solely related to the task environment. Similar approaches tried in the past have also relied on closed-loop response metrics[2,12]. Thus, in order to use the rule of thumb proposed here, one needs a method for predicting what this metric will be in a particular task situation. This author feels that progress has nevertheless been made, as there is a considerable body of manual control literature to provide guidelines for predicting, say, the gain-crossover frequency for various control tasks. In any case, the rule provided above implies that it is necessary to define this critical response variable only for the baseline (no-delay) tasks.

Because of the limited range of task environments explored in this study, additional study is required to determine the generality of the rule provided above; specifically, whether or not performance/delay tradeoffs are generally linear and, if so, to what extent the sensitivity varies with parameters of the task environment. The critical problem space remaining to be explored includes different tracking tasks (e.g., flight path regulation) and different types of inputs (continuous inputs, both command and disturbance).

Also to be explored is the extent to which the rule holds for different locations of the delay. The study reported here assumed the delay was in the innermost loop. For a delay located only in an outer loop (e.g., generation of the path error display), the rule would be applied by normalizing the added delay with respect to the crossover frequency appropriate to that particular loop.

Further application and validation of the general scheme should involve both model analysis to provide expectations concerning the generality of the rule of thumb and experimental work to validate the model predictions. To this end, further experimental research into the effects of simulator time delay might profitably include normalization of performance and delay as suggested here whenever the task is such that the desired closed-loop metrics can be computed.

Conclusions

Analysis performed with the optimal control model for pilot/vehicle systems showed that, for a limited range of task environments, relatively consistent effects of delay on performance could be obtained if both performance and delay were suitably normalized. Specifically, integral-squared error performance was normalized with respect to performance predicted for the baseline (no delay) task, and simulator delay was normalized with respect to the inverse of the gain-crossover frequency predicted for the baseline task. Normalization in this manner desensitized the predicted performance/delay relationship to both vehicle response parameters and performance requirements.

The following rule of thumb was computed for predicting the percentage increase in tracking error with simulator delay for step-command attitude-tracking tasks:

$$\% \text{ increase in ISE} = 90 * \omega_c * \Delta T$$

where ω_c is the crossover frequency (rad/sec) of the closed-loop system in the absence of simulator delay, and ΔT is the simulator delay in seconds.

Acknowledgement

This work was performed in part under contract to Charles River Analytics in support of Air Force Contract No. F33615-86-C-0532.

References

1. Smith, R.E., and Bailey, R.E., "Effect of Control System Delays on Fighter Flying Qualities", AGARD Conference Proceedings No. 333 "Criteria for Handling Qualities of Military Aircraft", 1982.

2. Allen, R.W., and DiMarco, R.J., "Effects of Transport Delays on Manual Control System Performance", Proc. Twentieth Annual Conference on Manual Control, NASA conference Publication 2341, 1984, pp 185-201.

3. Bailey, R.E., Knotts, L.H., Horowitz, S.J., and Malone, H.L. III, "Effect of time Delay on Manual Flight Control and Flying qualities During in-Flight and Ground-Based Simulation", AIAA Paper No. 87-2370, Proc. of the AIAA Flight Simulation Technologies Conference, Monterey, CA, August 17-19, 1987.

4. Baron, S., Muralidharan, R., and Kleinman, D.L., "Closed-Loop Models for Analyzing Engineering Requirements for Simulators", NASA CR 2965, February 1980.

5. Levison, W.H., and Papazian, B., "The Effects of Time Delay and Simulator Mode on Closed-Loop Pilot/Vehicle Performance: Model Analysis and Manned Simulation Results", AIAA Paper No. 87-2371, Proc. of the AIAA Flight Simulation Technologies Conference, Monterey, CA, August 17-19, 1987.

6. Monagan, S. J., Smith, R. E., Bailey, R.E., "Lateral Flying Qualities of Highly Augmented Fighter Aircraft", Technical Report AFWAL-TR-81-3171, Wright-Patterson Air Force Base, June 1982.

7. Ricard, G.L, and Puig, J.A., "Delay of Visual Feedback in Aircraft Simulators", Report No. NAVTRAEQUIPCEN TN-56, Naval Training Equipment Center, Orlando, FL, March 1977.

8. Riccio, G.E.; Cress, J.D.; and Johnson, W.V., "The Effects of Simulator Delays on the Acquisition of Flight Control Skills: Control of Heading and Altitude", Proc. Human Factors Society 31st Annual Meeting, 1987, pp 1286-1290.

9. Smith, R.E., "Effects of Control System Dynamics on Fighter Approach and Landing Longitudinal Flying Qualities", Technical Report AFFDL-TR-78-122, Vol. I, Wright-Patterson Air Force Base, Oh, March 1978.

10. Berry, D.T., Powers, B.G., Szalai, K.J., and Wilson, R.J., "A Summary of an In-Flight Evaluation of Control System Pure Time Delays During Landing Using the F-8 DFBW Airplane", AIAA Paper 80-1626, August 1980.

11. Smith, R.E., and Sarrafian, S.K., "Effect of Time Delay on Flying Qualities: an Update", J. Guidance, Control, and Dynamics , 9:578-584, Sept-Oct 1986.

12. Sinacori, J.B., "The Need for a Limited Theory of Perception and Action That can be Applied to Current and Future Issues for Visual and Motion Simulation", AIAA Paper No. 87-2297-CP, 1987.

13. Kleinman, D.L., Baron, S., and Levison, W.H., "A Control Theoretic Approach to Manned-Vehicle Systems Analysis", IEEE Trans. on Auto. Control, Vol. AC-16, pp. 824-833, No. 6, December 1971.

14. Levison, W.H., "The Optimal Control Model for the Human Operator: Theory, Validation, and Application," in Frazier, M.L., and Crombie, R.B., (eds): Proceedings of the Workshop on Flight Testing to Identify Pilot Workload and Pilot Dynamics, AFFTC-TR-82-5, Air Force Fiight Test Center, Edwards Air Force Base, CA, May 1982.

15. Baron, S. Lancraft, R. and Caglayan, A., "An Optimal Control Model Approach to the Design of Compensators for Simulator Delay", NASA Contractor Report 3604, NASA Ames Research Center, CA, 1982.

16. Levison, W.H., and Papazian, B., "Pre-Experiment Model Analysis in support of the NT-33A Ground Simulator Time Delay Study", Technical

Memorandum EPD-86-6, Bolt Beranek and Newman Inc., Cambridge, MA, April 11, 1986.

17. Baron, S., "Application of the Optimal Control Model to Assessment of Simulator Effectiveness", *Advances in Man-Machine Systems Research*, vol. 3, pp. 179-235. JAI Press Inc., 1987.

18. McRuer, D., Ashkenas, I., and Graham, D., "Aircraft Dynamics and Automatic Control", Princeton University Press, Princeton, NJ, 1973.

19. Baron, S. and Levison, W.H., "An Optimal Control Methodology for Analyzing the Effects of Display Parameters on Performance and Workload in Manual Flight Control," IEEE Trans. on Systems, Man and Cybernetics, Vol. SMC-5, No. 4, pp. 423-430, July 1975.

89-3272-CP

NEED-BASED EVALUATION
OF
SIMULATOR FORCE AND MOTION CUING DEVICES

Yorke J. Brown*
Cardullo, Brown and Associates
Binghamton, NY

Frank M. Cardullo*
State University of New York
Binghamton, NY

John B. Sinacori*
John B. Sinacori Associates
Pebble Beach, CA

Abstract. Since aircraft--and particularly military aircraft--can produce sustained periods of significant acceleration, whereas ground based simulators cannot, the technical problem of providing faithful force and motion cues in a simulator is particularly difficult. Many force and motion cuing strategies and devices have been proposed, and many have been tested, but opinion remains divided as to the effectiveness of, or even the need for, such equipment. We have developed a method for analyzing the dynamic properties of force and motion cuing in terms of the relationship between the pilot sensation of motion cues involved in typical aircraft maneuvers and the pilot sensation of the corresponding synthetic cues produced by a simulator equipped with force and motion cuing devices. This analytic method will serve to characterize the applicability of available devices, indicate areas of fruitful development of new technolgies, and guide research aimed at resolving applicability and usefulness issues. The analytic technique involves identifying the motions typical of operational flight maneuvers and then characterizing the pilot's sensory response to the resulting accelerations and forces in the frequency domain. When a corresponding spectrum for the sensation of synthetic stimuli is overlaid, quantitative comparisons may be made. The analysis effectively integrates the relevant effects of mission profile, aircraft response, pilot sensory system response, and simulator cuing device performance into a unified graphical presentation.

INTRODUCTION

The problem of designing a flight simulator which can provide realistic motion cues to the pilot has plagued the industry from its very earliest days. Ed Link's great accomplishment in developing the original "Link Aviation Trainer No. 1" in 1924 lay in the use of pneumatics to provide three degrees of motion freedom in direct response to cockpit flight control inputs. The trainer, which was intended for primary flight training, allowed the student to see the effects of control inputs as the cockpit assembly reproduced aircraft attitude changes in roll, pitch and yaw. It was fortunate for Link, however, that visual motion perception can sometimes dominate somatosensory and vestibular perceptions, because the motions of the trainer, although nearly correct visually, provided largely incorrect force and motion cues. When the Army, in 1936, began using Link's famous "Blue Box" for instrument training, the problems became more apparent: the pivot point was below the cockpit rather than at the simulated aircraft CG, and the simulator roll motions made even the most perfectly coordinated turn feel like a slip. In spite of the tremendous contribution that Link Trainers were making to pilot training, by the end of the war the government was specifying instrument trainers without motion systems.

Even today, the modern six-post motion system receives mixed reviews from users. No motion system can possibly reproduce the large sustained accelerations produced by combat aircraft, so just in the regime where force and motion cuing becomes dramatically significant to aircrew performance and training, elab o rate motion platform technology seems to fail. Although motion platforms have proven their worth in helicopter simulators and in some transport simulators, the Air Force still prefers to avoid them for fighters. The F-16 simulator, for example, has no platform motion system.

Barring an unforseen revolution in the technology of force and motion cuing, it is evident that it is hopeless to attempt to provide realistic force and motion stimuli in the sense that the acceleration forces produced by the aircraft can be replicated in the simulator. Sustained acceleration is not possible without sustained displacement, and any attempt to simply apply whole body acceleration forces directly will always require inappropriate counter forces. The key to circumventing this dilemma is to recognize that what matters in simulation is not the reality of the force and motion stimuli, but the sensations--and, more specifically, the perceptions--associated with force and motion. The simulator is not called upon to replicate the motions of the aircraft, but rather to replicate the aircrew's perceptions associated with that motion. Current practice in motion cuing with hydraulic platforms utilizes this approach to some extent by providing critical onset cues followed by subliminal washout, and by using "gravity align" platform attitudes to simulate

* Member, AIAA

some aspects of sustained acceleration. Nevertheless, motion platforms use actual acceleration to stimulate the pilot's sensation of acceleration; a more direct example of the synthetic stimilus technique is the g-seat--such as that used in the F-16 simulator. The g-seat provides no acceleration and exerts no forces, but rather simulates the sensation of g-induced buttocks and back pressure through changes in seat area and firmness. The device stimulates acceleration-induced postural changes by adjusting seat orientation. The result is a psychological suggestion to the pilot of acceleration forces, which he interprets as due to aircraft motion.

In pursuing this avenue of simulator design, two related questions emerge. The first is that of the available techniques for providing stimuli which can be interpreted by the human perceptual systems as force and motion cues. The second is that of the relevance of each of the various stimuli which are present in the real-world environment. Clearly, if a great deal of effort is to be expended in searching for and implementing force and motion analogs or stimuli which produce sensations similar to those due to acceleration, the designer will be interested in identifying only those which are relevant to aircrew performance and training. In order to characterize candidate techniques on the basis of need, the designer must analyze the entire stimulus-response chain from the original aircraft maneuver through the physical stimuli provided to the pilot, through the physical responses of the human sensory receptors which detect the stimuli, and finally to the perceptual effects which elicit the behaviors resulting in the pilot's specific performance. All possible motions of the aircraft are not necessarily relevant to simulation: some areas of the performance envelope of the aircraft are never exercised; some stimuli are not detected by the human; and some perceptions are not relevant to performance. A critical task of the simulator designer is to identify those regimes of motion, sensation, perception, and behavior which are relevant to pilot performance, and then to devise techniques for producing--by synthetic stimuli--the appropriate range of pilot perceptions.

There currently exists no unified approach to the quantitative analysis of flight simulator cuing requirements. As part of an effort to develop new techniques for synthetic cuing we have developed an analytic method which graphically shows the relationships between the pilot's sensations due to aircraft motion and the pilot's sensations due to simulator motion cuing devices. The approach does not address directly the issues of the perceptual consequences of synthetic versus real sensory stimuli, but proceeds on the dual assumption that, given enough sensory information, the human nervous system will will provide a credible perception of motion.

The approach presented here elaborates the method described by Cardullo and Sinacori (1). It includes a specialized analysis of the fighter pilot's actual force and motion environment and the sensory processing of the resulting stimuli. The results are presented in the form of spectral density charts familiar from the theory of system analysis. A similar analysis of the stimuli and perceptual processing of synthetic cues allows for direct comparison with the real world case. This comparison is the basis of a quantitative evaluation of the need for cuing in certain regimes and of the effectiveness of candidate cuing devices.

HUMAN PERCEPTION OF MOTION

Humans do not directly perceive the nature of their surroundings or their motion through it. A person's concept of his surroundings and of his own motion is built up by the central nervous system at various levels of consciousness by synthesizing the nervous signals derived from a wide variety of sensory organs. Although the eye, for example, is physically much like a photographic camera, it does not, in any real sense, send pictures to the brain. Rather, the brain uses the signals on the optic nerves to infer a concept of the physical space around the subject. We experience this concept as a visual image--but an image quite different in character from those captured by the retinas. Even persons with some types of gross optical defects can learn to form a complete, continuous visual image; but persons with no visual experience of certain surroundings find it impossible to organize the visual sensations into a meaningful perception of the space.

Although intimately connected, the processes of sensation and perception are quite distinct. Sensation is the process by which a sensory organ responds to a stimulus and sends a signal to the central nervous system. Perception is the integrative process of by which the CNS builds up a concept of bcdy state and surroundings; it is the process by which we give meaning to sensation. Physically, the response of a sensory organ to stimulation is to change its rate of neural discharge. Consequently the afferent firing rate may be considered a signal indicating the strength of stimulation. The quality of stimulation--flavor, odor, color, coldness, hardness--is an interpretation made by the perceptual system based on combinations of signals received from different sensory end organs. Each sensory organ itself can properly be considered a transducer subject to the same kinds of analytical treatment used on artificial transducers.

Perception is vastly more complex than sensation, and not well understood in any detail. Nevertheless it is clear that the perceptual system is remarkably flexible: it can develop the same accurate percept based on different sets of sensations--even if the sensory data is not entirely self-consistent. Similarly, the brain can construct an illusory percept from sensory data which, whether by accident or design, is consistent with that percept--even if incomplete. Visual illusions of fictitious spaces, or simply illusions of self motion, are particularly easy to create through pictures, cinema, video--and simulator visual systems. Faithful replication of all the sensations due to aircraft motion is not possible in a simulator, but because of the inherent flexibility of the perceptual process, useable replication of the perception of aircraft motion is possible if enough artificially produced sensory data are provided to the brain.

The most direct perceptions of self-motion derive from sensations received by the visual and vestibular systems. Although the visual system generally dominates the vestibular system in pure motion perception, the two types of sensory data are largely compelementary. Vision is highly developed for detecting position, attitude, and slow changes in these; the labyrinth is highly developed for detecting both angular and linear acceleration, particularly at high frequencies. Both complement each other. For example, vision improves the accuracy of integration of velocity and acceleration signals received by the semicircular canals and otoliths, while the labyrinth provides the signals necessary for inertial stabilization of the eyes during head movement.

Humans also infer motion from other somatosensory inputs. A particularly dramatic example is the limb and head loading a pilot feels during acceleration. Numerous other sensations also contribute to the sensation of motion: acceleration forces produce pressure on supported body parts; clothing and equipment change weight and shift on the body; the buttocks, back, and elbows scrub against the seat; internal organs are compressed and shifted resulting in an impressive variety of physiological consequences; and the pilot's attempts at motion are affected both subtly and grossly by the acceleration force field.

All these effects synergize one another. Interestingly, aviation is sufficiently alien to humans that many combinatioins of the sensations associated with flying seem contradictory to the pilot, leading to illusions, vertigo, disorientation, and airsickness. Simulator cuing devices may take advantage of all these channels of stimulation to help build in the pilot an integrated sensation of motion similar to that experienced in the real world despite the fundamental limitation that no sustained acceleration is possible. An optimum design, however, requires that the perceptions occuring in actual flight be well matched to the perceptions produced by the simulation equipment.

THE CHARACTERISTICS OF AIRCRAFT MOTION

Aircraft--particularly modern, high performance combat aircraft--are capable of motions which are never used in actual flight. Some regions of the envelope are simply declared too dangerous and are therefore prohibited to the pilot; others are not tactically or operationally useful. Simulators, therefore, need not be designed to provide cues associated with all the possible motions of the aircraft--only with those motions which are actually undergone, which can be perceived by the human sensory system, and which are relevant to performance or training.

Even in combat, pilots tend to fly specific, fairly well defined maneuvers. A particular engagement is a particular sequence of these standard maneuvers, selected by the pilot as the developing tactical situation warrants. There are many exceptions, variations, and adjustments, of course, but generally a pilot can describe an engagement or a mission in terms of named maneuvers. Furthermore, each maneuver tends to be constructed of brief periods of acceleration onset, steady acceleration, and accleleration offset. This type of acceleration profile gives the pilot periods of relatively steady conditions which minimize his physical stress and which give him opportunities to assess the tactical situation. Figure 1 shows a graph of the vertical acceleration (Gz) during a hypothetical maneuver of this sort. The pilot generally maneuvers at +Gz in order to bring about a change in state of the aircraft, and then he "unloads." If the "pull" is perfectly and smoothly executed, then the curved portions of the profile may be approximated as segments of sinusoids. Although we have made no empirical studies of actual aircraft flight data, a detailed theoretical study of the pop-up ground attack mission of the F-4 confirms this assertion.

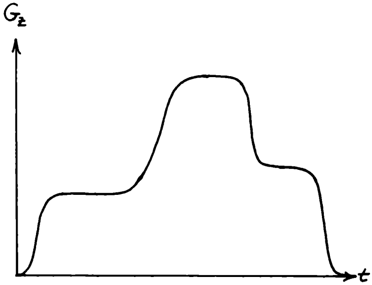


Figure 1. Segment of a time history, f(t), of an idealized combat mission

The Fourier transform of a time-dependent function composed of sinusoidal-edged pulses is a complicated sum of sinusoids under an envelope with a characteristic $1/\omega + \omega/(\omega_0^2 - \omega^2)$ dependence. The result is a function with a $1/\omega$ cutoff at twice the maximum frequency of the edge-forming sinusoids. The zero-frequency maximum is just the total g-exposure--the time integral of the acceleration curve.

To begin a hypothetical illustrative example, suppose that the mission we wish to simulate involves pulls to 6G in 1 second. This pull constitutes a change of 5G in 1 second, or 2.5G in one quarter of a cycle. The characteristic frequency is therefore 0.5 Hz or 3.1 rad/sec. The cutoff frequency is 1 Hz or 6.3 rad/sec. The maximum G-onset rate is 7.9 G/sec. The Fourier transform of the sum of many such missions, normalized for mission length, is shown in figure 2. The mixture of many pulses of Gz washes out all the structure, leaving only the characteristic 1 Hz cutoff. This function, $F(\omega)$ is a representation of the aircraft Gz in the frequency domain.

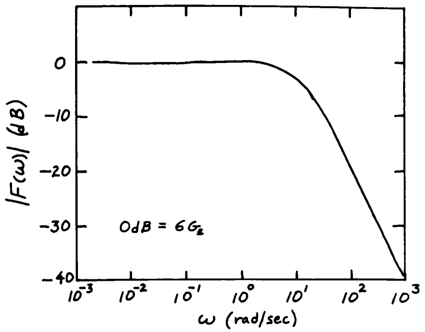


Figure 2. The frequency domain representation of the vertical acceleration of the aircraft in the hypothetical combat mission.

Similar analysis holds for other parameters describing aircraft motion. For the purpose of simulator force and motion cuing analysis, however, only four parameters need to be considered--provided that only coordinated flight is considered. These are vertical acceleration (Gz), pitch rate (q), longitudinal acceleration or thrust (Gx), and roll rate (p). Any other parameter may be derived from these, while this set provides the most direct coupling to human motion sensing channels.

ANALYSIS OF SENSORY PROCESSING

If a sensory organ is viewed as a transducer which converts sensory stimuli into a neural signal, then the entire process of sensation, including the mechanisms of stimulus, may be analyzed using the classical tools of signal processing. In a linear system, the only requirements for faithful transmission of a signal are that the transmission channel have constant gain and a phase shift proportional to frequency. If these requirements are not met, the characteristics of the channel bandpass will be conferred on the signal.

Figure 3 shows a simplified block diagram of the sensory signal path from aircraft to afferent nerve fibers for both the real world and simulated cases. The only significant difference between the two cases is that, in the simulated world, the aircraft motion signals must pass through a motion cuing device, whereas in the real world it is the aircraft itself which transforms the aircraft motion to physical stimuli. For the sake of simplicity, we will take the aircraft to be a unity-gain channel, although, in fact, minor channel shaping is induced by the dynamics of the aircraft seat and other factors. The frequency domain representation of the aircraft motion, whether simulated or real, is $F(\omega)$.

The sense organs each have their own responses, characterized by transfer functions $T(\omega)$, based on physical models. Models are available for all the principal organs of force and motion perception (2, 3, 4). Although these organs are not always linear in their responses, linear approximations make a good starting place for this type of analysis. As examples, the magnitude frequency response of the otoliths, based on a model due to Fernandez and Goldberg (5), and a magnitude frequency response for the Pacinian corpuscles, based on a model due to Borah (3), is shown in figure 4. The otoliths sense linear acceleration; the Pacinian corpuscles sense deep tissue pressure such as that exerted on the buttocks of a seated person. The plots are normalized so that zero dB is the gain at maximum sensitivity. In both cases, the input is the specific stimulus and the output is afferent firing rate (AFR). The frequency-domain representaion of the AFR in the real world case is called $A(\omega)$; in the simulated case it is called $S(\omega)$.

The AFR for each sensor in each case--real or simulated--is simply given by

$$A(\omega) = T(\omega)\ F(\omega)$$

$$S(\omega) = M(\omega)\ T(\omega)\ F(\omega)\ ,$$

where $M(\omega)$ is the transfer function of the cuing device stimulating a particualar type of sense organ.

In terms of sensation in a specific channel, the fidelity of the simulation is characterized by the extent to which $A(\omega)$ and $S(\omega)$ are similar. In terms of perception, however, the fidelity of the simulation is determined by the details of the sensory integration processes. The perceptual system can use the sensory information available from all channels to infer a percept of motion. The aim of the simulator designer should be to "mix and match" cuing devices to produce a suitable collection of $S(\omega)$s which match the collection of $A(\omega)$s determined by the mission.

AN EXAMPLE

As an illustrative example of the technique, let us consider the problem of Gz cuing in the hypothetical aircraft whose Gz spectrum, $F(\omega)$, is given in figure 2. Vertical acceleration is cued in many simulators with a motion platform, the primary function of which is to stimulate the otoliths. The transfer function, $T(\omega)$, for the otoliths is given in figure 4.

The real-world spectrum of the otolith output signal is $A(\omega)=T(\omega)F(\omega)$, as shown by the upper solid curve in figure 6.

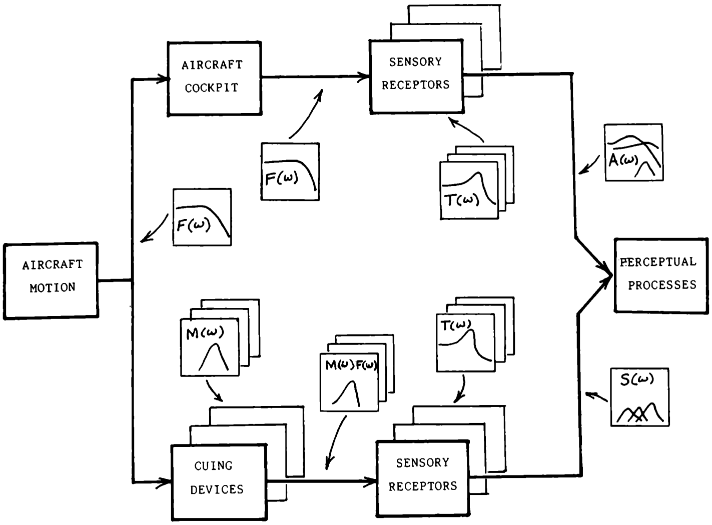


Figure 3. Block diagram of the signal processing involved in sensation of aircraft motion and simulator motion cues.

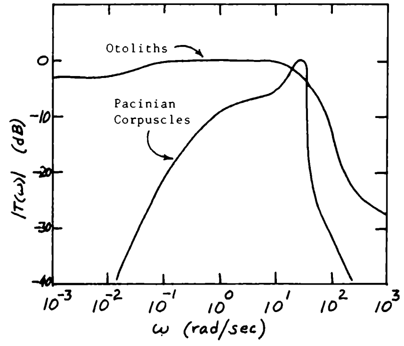


Figure 4. The magnitude of the transfer functions of the otoliths and the Pacinian corpuscles.

In the simulator, the otolith sensory channel includes the filtering action of the motion platform. A hypothetical, high-performance motion platform would behave like a slightly underdamped, second order system with a cutoff frequency of about 3 Hz (19 rad/sec). At high frequencies, as the frequency increases, the excursion produced by a constant amplitude acceleration input decreases. At low frequencies, however, the exursion amplitude increases with frequency and will exceed the capacity of the platform if not attenuated. Thus, in the high frequency regime, the motion system acceleration frequency response is just the same as its displacement frequency response; but at low frequencies, because of the limited excursion of the platform, its acceleration response must by attenuated by the cuing algorithm at a rate of ω^2/ω_m^2. Here, ω_m is the low frequency cutoff, and is given by the ratio of maximum displacement to maximum acceleration: ω_m^2 =Amax/Xmax.

For the example, suppose that the maximum excursion amplitude is 3 feet and the maximum acceleration capability is 2G. Since we need to cue the onset of a 5G pull, the cues must be scaled by Ko=2/5. The low frequency cutoff will be at 4.6 rad/sec (0.74 Hz), with amplitude rising at 40 dB/decade. The frequency response of the hypothetical motion system is shown in

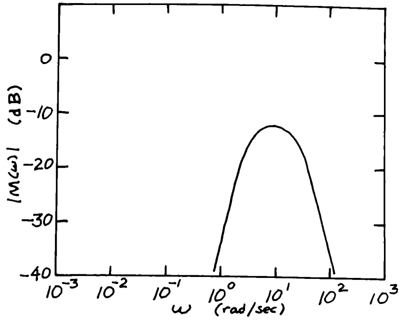


Figure 5. The transfer function of a hypothetical motion platform.

figure 5. The platform obviously does not have the low frequency response to stimulate the otoliths in the same way as the real aircraft.

The otolith AFR, S(ω)=F(ω)T(ω)M(ω) in the simulator, is shown as the lower solid curve in figure 6. The comparison of A(ω) and S(ω) is evident in this diagram. It reveals that the motion platform provides, in this hypothetical example, only about one third of the desired stimulus to the otoliths at its most effective frequency. It fails entirely at low frequencies, but replicates the high-frequency regime fairly well.

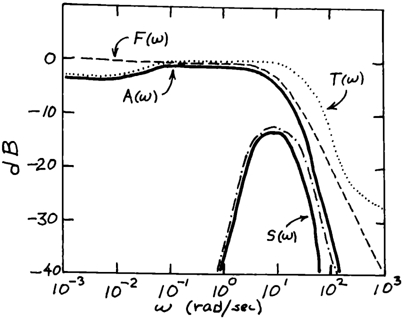


Figure 6. Summary of the analysis of the otolith sensory channel.

The question now is, given that the pilot senses (and perceives) low frequency acceleration (as shown in the A(ω) plot), can we find another cuing device to fill in some more of the spectrum? The approach must be to devise a cuing device to excite a different sensory channel at low frequencies; low frequency stimulation of the otoliths is probably a lost cause short of direct neural stimulation or putting the simulator cockpit on a centrifuge. A g-seat is a possible choice.

The g-seat stimulates the pressure sensors in the buttocks by varying the firmness of the seat cushion as a function of Gz. The result is that the effective seat area is reduced with increasing Gz, thus increasing the compression of tissue between the ischial tuberosities and the seat pan. (The g-seat also raises and lowers the pilot thus stimulating back-scrubbing and vision, but we ignore these effects for the sake of the

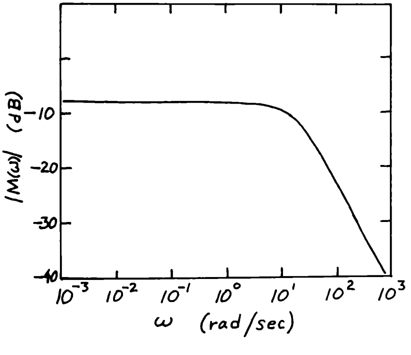


Figure 7. The transfer function of a hypothetical g-seat.

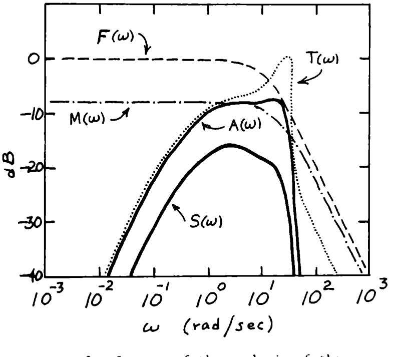


Figure 8. Summary of the analysis of the buttocks pressure receptor sensory channel.

example.) The amount of seat area decrease is only about a factor of 1/3, so cue scaling will be required.

The sensory organs stimulated by buttocks pressure are primarily the Pacinian corpuscles. Their transfer function, $T(\omega)$, for pressure to AFR, is given in figure 4. The afferent firing rate due to aircraft acceleration is $A(\omega)=T(\omega)F(\omega)$, shown as the upper solid curve in figure 8.

In the simulator, the commanded acceleration, $F(\omega)$, acts on the pilot through the g-seat dynamics rather than the aircraft seat dynamics. In the aircraft, the seat transfer function is simply the constant ratio of pilot mass to seat area, m/a, but in the g-seat, the applied acceleration is always Gz=1 (at low frequencies, anyway), and F(W) acts on seat area. Practically, very good performance would be a single-pole low pass characteristic with a cutoff of about 3 Hz and a dc gain given by the amount the seat area can be reduced. If the seat area can be reduced by a factor of about 1/3, then the maximum pressure which can be exerted is that equivalent to 3G (a 2G increase). For the hypothetical mission, maximum acceleration is 5G, so the gain of the seat is Ko=2/5. The frequency response of the hypothetical g-seat, $M(\omega)$, is shown in figure 7.

The final sensory output is $S(\omega)=F(\omega)T(\omega)M(\omega)$ in the simulator, and is shown as the lower solid curve in figure 8. In terms of frequency response, the g-seat is quite faithful to the real world--its cues are simply weaker, as expected.

The purpose of adding a g-seat to the hypothetical simulator design was to extend the bandwidth of vertical acceleration perception to lower frequencies. Figure 9 presents a summary of the motion platform and g-seat sensory outputs, together with the real world otolith and pressure receptor channels. The g-seat extends the cuing bandwidth down to about 0.3 rad/sec (0.05 Hz, or about 10 seconds of steady Gz cuing), although the cues are still some 8 dB weaker than those received in the real world.

CONCLUSION

In the modern aviation environment, Ed Link's emphasis on providing realistic cues is an unachievable--and unnecessary--goal. What is required is not fidelity to the aircraft, but rather fidelity to the pilot's perceptions of the piloting task. From this perspective, synthetic cues are just as useful as replications of the actual cues.

The technique described here can certainly be useful to the simulator designer in selecting an optimum configuration of simulator cuing devices, but it also has an important use in guiding research on new cuing equipment. There is a need, for example, to develop cuing devices for the sustained high Gz flight regime. Currently available devices, like the g-seat, offer inadequate bandwidth and inadequate gain, and they fail to stimulate all the available sensory channels. One promising avenue for stimulating the cardiovascular response to high Gz is lower body negative pressure (LBNP). Perhaps the buttocks pressure sensors can be further stimulated or their sensitivity enhanced by heating or cooling. Limb, head, and equipment loaders might be useful in stimulating the proprioceptive system. Perhaps low cost peripheral vision motion stimulators can provide much of the benefit of a wide field of view visual system without its enormous attendant cost. Even direct neural stimulation is no longer out of the question. Certainly, empirical research will be necessary to determine the perceptual characteristics and the usefulness of combinations of these devices. The starting point of any such research should be an evaluation of the relationships between the sensations produced by the proposed cuing devices and those stimulated by actual aircraft motions.

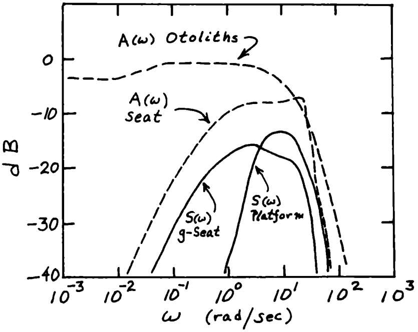


Figure 9. A summary diagram showing the two channels of Gz sensation studied.

ACKNOWLEDGEMENTS

We are very grateful to Ed Billman, Grant McMillan, Ed Martin, and Gary Riccio for numerous fruitful discussions of both the problem of characterizing cuing devices and this method of doing so.

REFERENCES

1. FM Cardullo, JB Sinacori, "A broader view of in cockpit motion and force cuing," Proceedings of 1988 AIAA Flight Simulation Technologies Conference, Atlanta, GA (1988).

2. DR Gum, "Modeling of the human force and motion sensing mechanisms," AFHRL-TR-72-54 (1973).

3. J Borah, LR Young, RE Curry, "Sensory mechanism modeling," AFHRL-TR-77-70 (1977).

4. C Ormsby, LR Young, "Perception of static orientation in a constant gravitoinertial environment," Aviat. Space and Environ. Med., 47, 159 (1976).

5. JM Goldberg, C Fernandez, "Physiology of peripheral neurons innervating otolith organs of the squirrel monkey I, II, III," J. Neurophysiol. 39, 970 (1976).

89-3274-CP

THE USE OF VESTIBULAR MODELS FOR DESIGN AND EVALUATION OF FLIGHT SIMULATOR MOTION

Steven R. Bussolari, Ph.D.* and Laurence R. Young, Sc.D.**
Man-Vehicle Laboratory†
Massachusetts Institute of Technology
Cambridge, Massachusetts

Alfred T. Lee, Ph.D.
NASA Ames Research Center
Moffett Field, California

Abstract

Quantitative models for the dynamics of the human vestibular system have been applied to the design and evaluation of flight simulator platform motion. These vestibular models have been used to evaluate the motion fidelity of a transport category aircraft (Boeing 727) simulator in a pilot performance and simulator acceptability study. In addition, an optimal simulator motion control algorithm has been generated to minimize the vector difference between sensed spatial orientation estimated in flight and in simulation. The optimal motion controller has been implemented on the motion system of the Vertical Motion Simulator NASA Ames Research Center and evaluated experimentally through measurement of pilot performance and subjective rating during VTOL aircraft simulation.

Introduction

The use of flight simulation as a tool for pilot training, certification, and flight control system development has increased dramatically in recent years as the associated technology has become more sophisticated. To the extent that the pilot makes use of visual, aural, tactile, and motion cues in aircraft control it is necessary to reproduce those cues accurately in the simulator. A fundamental question concerns how much engineering and psychological fidelity is necessary to produce the same piloting behavior in the simulator as that observed or expected in the actual aircraft. It is recognized that the fidelity requirements of simulators used for flight training will, in general, differ from those used for research. Once those requirements have been established, however, the simulator designer must decide how best to generate the cues that enhance fidelity.

Under current regulations of the Federal Aviation Administration, all simulators used for civil aircraft training within the U.S. are required to provide at least three degrees of freedom (DOF) of platform motion. Simulators used for initial, transition, and upgrade training and checking are required to have full six DOF platform motion systems. The requirement for platform motion is ostensibly based on the assumption that physical fidelity is highly correlated with training effectiveness. Since aircraft are capable of motion in all six axes (three translational, three angular), it is believed that the absence of motion in the simulator would significantly reduce its training effectiveness. Although no data exist to confirm or contradict this hypothesis for civil transport operations, training transfer studies conducted on general aviation and military training simulators do not support this assertion.[1] While it is arguable that the motion systems in these studies were of the highest quality, the absence of motion effects across such diverse training environments and simulator equipment considerably weakens the case for requiring elaborate motion platform systems in flight simulators used for training pilots in fixed wing aircraft operations. Simulator motion requirements for hovering aircraft have yet to be firmly established.

The purpose of this paper is to describe work directed towards understanding the influence of motion platform systems on pilot behavior. First, the development of a quantitative model of human spatial orientation is outlined. The model provides the system designer with a tool by which motion platforms and their associated drive logic can be developed. an optimal simulator motion control algorithm that uses the spatial orientation model is presented along with the results of an experimental evaluation of the design technique in a VTOL aircraft simulation. Finally, the results of a recent study which used the vestibular models to evaluate the influence of platform motion variations on pilot performance and ratings of simulator fidelity in a transport category aircraft (Boeing 727) simulation are presented.

The Vestibular Model

A substantial portion of our research has been the development of quantitative models for human spatial orientation based primarily upon physiological models of the human vestibular system. These models relate linear acceleration to perception of tilt and linear motion and angular acceleration to perception of angular velocity.[2,3,4] Early in this process we recognized the relationship between spatial orientation models and the influence of flight simulator and aircraft motion on pilot orientation perception and performance. Our early work tying semicircular canal and otolith models to fixed base versus motion base simulation[5,6,7] indicated the relationship of motion cues to the development of pilot lead, particularly in the task of flying aircraft with marginal stability.

* Assistant Professor, Member AIAA

** Professor, Member AIAA

† Department of Aeronautics and Astronautics

Simulator motion fidelity can be expressed quantitatively by comparing the modeled perceptions of the simulator pilot with those of the pilot of the actual aircraft in flight (see Figure 1). The physical motion of the aircraft and that of the simulator are used as inputs to identical human spatial orientation models; one of the aircraft pilot and one of the simulator pilot. The outputs of these models are taken as the motion perceptions, respectively, of the aircraft pilot. The degree to which the simulator effectively reproduces, in the simulator pilot, the perception of actual aircraft motion is expressed as the difference of the outputs: the spatial orientation error. The use of a linear model to produce spatial orientation error as a quantitative fidelity metric is applicable to all types of motion drivers, including non-linear, adaptive washout systems.

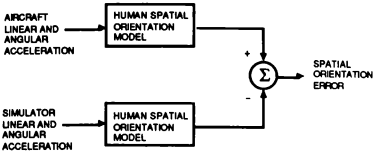

AIRCRAFT LINEAR AND ANGULAR ACCELERATION
HUMAN SPATIAL ORIENTATION MODEL
+
Σ
SPATIAL ORIENTATION ERROR
-
SIMULATOR LINEAR AND ANGULAR ACCELERATION
HUMAN SPATIAL ORIENTATION MODEL

Figure 1. The comparison of simulator platform motion with aircraft motion through the use of spatial orientation models.

Rather than use models based solely upon the mechanics of the vestibular end organ mechanics[8,9]; or upon the transfer function to perception of spatial orientation[3,10]; we choose to employ models whose output reflects the typical firing rate of the vestibular afferent neuron. The rationale for this approach is that vestibular afferent firing rate is a more relevant indication of the perception of spatial orientation because the dynamics of the transduction of mechanical events to electrical impulses in the peripheral nervous system are included. There is, of course, additional signal processing (with associated dynamics) that takes place at higher levels of the central nervous system to produce the human's perception of motion, but validated models for this have yet to be formulated. The model parameters were chosen based upon the squirrel monkey experiments of Fernandez and Goldberg [11,12] and are assumed to be a reasonable approximation of those of the human. A block diagram of the spatial orientation model is depicted in Figure 2 for linear acceleration on the longitudinal axis (surge) and angular velocity in pitch. Similar models are constructed for motion in the other two linear and two angular axes.

Angular motion in pitch is sensed by the semicircular canal as angular acceleration and by the otolith as an apparent change in body force due to the angular displacement of the gravity vector. Actually, the coupling of pitch attitude to the otolith varies as the sine of pitch angle; therefore, we make the small pitch angle approximation to preserve the linearity of our model. The coupling between the perceived linear acceleration and pitch angle as sensed by the otolith organ is exploited in flight simulator motion design by tilting the simulator cab (g-tilt) to create the illusion of sustained linear acceleration.

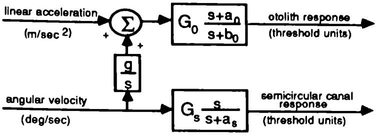


Figure 2. Block diagram of spatial orientation model in pitch-surge direction. Otolith model parameters: G_0=5.32 sec^2/m; a_0=.076 rad/sec; b_0=.190 rad/sec. Semicircular canal model parameters: G_s=.659 sec/deg; a_s=.169 rad/sec. (g is gravitational constant).

The gain of each model was adjusted so that its output is an average normalized firing rate of the vestibular afferent neuron expressed in threshold units. The normalizing factor (one threshold unit) corresponds to the level of angular velocity or linear acceleration that is just perceptible to a pilot performing fight tasks in a simulator[13]. Simplification of the model was achieved by eliminating dynamics (e.g. the .003 sec. short lag time constant of the semicircular canals) that were well outside the capabilities of existing flight simulator motion systems.

The Optimal Washout System

Design

Given the linear vestibular model as a tool, we are able to address the design of flight simulator motion drive logic as an optimal control problem[14]. The structure of the optimal washout system is shown in Figure 3.

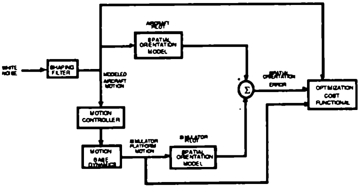

WHITE NOISE
SHAPING FILTER
MODELED AIRCRAFT MOTION
AIRCRAFT PILOT
SPATIAL ORIENTATION MODEL
MOTION CONTROLLER
MOTION BASE DYNAMICS
SIMULATOR PLATFORM MOTION
SIMULATOR PILOT
SPATIAL ORIENTATION MODEL
Σ
SPATIAL ORIENTATION ERROR
OPTIMIZATION COST FUNCTIONAL

Figure 3. Optimal washout system structure.

The basis of the design technique is the assumption that the utility and acceptability of flight simulator motion is optimized by minimizing the expected value of the mean-square spatial orientation error as computed by the model given the physical constraints of the simulator motion base. A weighted quadratic optimization cost functional is formed by combining the state vector of the spatial orientation model with the state vectors of the aircraft model and the simulator dynamics. The elements of the weighing matrix are assigned *a priori* based upon the desired maximum values of spatial orientation error and simulator motion platform acceleration and travel. Aircraft motions resulting from pilot inputs and external disturbances are modeled as a random process with a rational spectral density that can be adjusted by means of a shaping filter to match the particular aircraft and range of expected flight tasks.[15] The motion controller is then synthesized off-line using standard linear-quadratic-Gaussian optimal control methods.[16]

Validation of Optimal Washout System

In an attempt to validate our model, we have used the optimal control design technique to produce a motion washout system for two of the six degrees of freedom (pitch and surge) of the Vertical Motion Simulator (VMS) at NASA Ames Research Center . This facility, designed primarily for VTOL simulations, has a total vertical travel capability of 14m and a total horizontal travel of 9m. For the purpose of the validation study, the simulator cab was rotated 90 degrees so that the longitudinal axis of the cockpit was aligned with the horizontal trackof the motion base, providing a full 9m of travel in the surge direction. The aircraft mathematical model selected for the experimental evaluation was that of a vectored-thrust VTOL vehicle. The vehicle's transfer function in the pitch-surge direction between cyclic pitch input and aircraft longitudinal direction was fifth order and included parameters that modeled aerodynamic drag, pitch rate damping, and control system lag.

In order to assess the sensitivity of our motion drive logic to its design parameters and compare the experimental system with an established washout, six different washout systems were developed. Three versions of the optimal washout system (OWS) were synthesized by choosing different values for the weights in the quadratic cost functional. The first, OWS Nominal, was designed to make maximum use of the simulator motion base travel and placed equal weight on the modeled otolith and semicircular canal orientation errors, relative to their thresholds. The second, OSW Decreased Gain, was generated by placing large weights on platform motion states as compared to the computed orientation error. This washout was designed to make use of approximately half of the VMS platform horizontal travel. The third, OWS High Otolith Weighting, was synthesized by placing twice the weight on the orientation error contribution of the modeled otolith response as that placed on the orientation error contributed by the semicircular canals. In addition to the three motion drive systems synthesized by the optimization technique, three versions of the motion controller currently used with the VMS were implemented in the pitch-surge axes. All three were of the crossfeed type as designed by R. Bray at NASA Ames Research Center and described by Sinacori.[17] The first, Ames Nominal, was tuned by the designer to make maximum use of the simulator motion travel given the types of flight maneuvers anticipated in the study. The second, Ames Decreased Gain, was modified to reduce the horizontal travel of the simulator cab by a factor of two by reducing the gain of the linear washout filter. The third, Ames Increased Omega, was generated by increasing the break frequency (omega) of the high-pass washout filter in order to decrease the low-frequency content of the simulator motion. This has the effect of decreasing the travel requirements of the simulator without attenuating the amplitude of the high frequency motion.

The response of each of the six motion washout systems as implemented on the VMS are compared in Figure 4 for a single dash-quick-stop maneuver. In this maneuver, the aircraft is pitched nose-down to accelerate forward to a given velocity and then pitched nose-up to decelerate rapidly to a stationary hover.

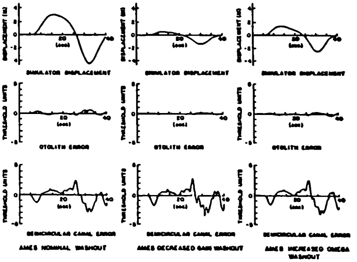


Figure 4a. Response of Ames crossfeed washout system to a single dash-quick-stop maneuver.

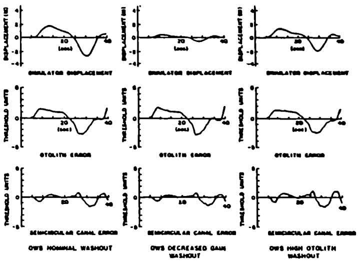


Figure 4b. Response of optimal washout system (OWS) to a single dash-quick-stop maneuver.

Identical pilot inputs were given to the aircraft mathematical model for each of the six washout systems. For each motion drive system, the measured simulator displacement, computed otolith error, and computed semicircular canal error are plotted versus time. The time responses of the Ames washouts are characterized by extremely low otolith error due to the fact that this washout system placed emphasis on the coordination of the tilt of the simulator cab with longitudinal acceleration. The reduction of washout filter gain in the Ames Decreased Gain case produces the predicted effect of a lower simulator horizontal displacement. This is achieved at the expense of a slight increase in the computed semicircular canal error. A similar effect can be seen for the Ames Increased Omega washout. The OWS Nominal washout was generated with equal weighting on otolith and semicircular canal error in the cost functional and this is reflected in the balance between the two errors for the dash-quick-stop maneuver. The OWS Decreased Gain washout commands a smaller simulator displacement with very little change in the modeled otolith and semicircular canal error. The OWS High Otolith Weighting washout, designed by placing high weights on the computed otolith error, generates otolith error that is only slightly less than that produced by the OWS Nominal washout. This, combined with the observation that the computed otolith and semicircular canal errors do not change significantly when the simulator platform travel is reduced by the OWS Decreased Gain washout, indicates the existence of an apparent insensitivity of the optimization equations with respect to the orientation error. The insensitivity of optimal controller performance to changes in the weights of certain components of its cost functional is commonly encountered in optimal controller design[16] and may be compensated for by placing larger weights on those components and their time derivatives.

VTOL Simulation Methods

In order to determine the effect of each of the six motion washout systems upon pilot performance and simulator acceptability, a series of evaluation experiments were performed using the VMS. Four NASA test pilots participated in the study, all of whom were current in VTOL aircraft (three had extensive experience in the VMS). Each pilot subject was given a simulator familiarization and practice session followed by two experimental sessions of approximately 6 minutes each. During each experimental session, the pilot subject experienced four motion conditions: fixed base and three of the six washouts described above. The order of presentation of the motion conditions was adjusted for each subject so that the effect of learning could be assessed. The small number of subjects tested precluded a complete counterbalancing of all motion conditions to compensate for learning effects; however, learning proved not to be a significant factor in this study. In each case that the motion base was active, all six degrees of motion freedom were used; however, only motion control in the pitch and surge axes were manipulated. The other four motion axes were controlled by the Ames Nominal washout throughout the experiment. After a period of warm-up flight, each pilot performed a formation flight task in which a lead VTOL aircraft was placed in the visual scene 33 meters in front and slightly to the left of the simulator. The pilot was instructed to maintain his aircraft in formation at a fixed distance from the lead aircraft. During each 75-second trial, the lead aircraft (with flight characteristics identical to the simulated VTOL aircraft piloted by the subject) was subjected to a pseudo-random pitch disturbance produced by a sum of five sinusoids of equal amplitude at frequencies of 0.257, 0.513, 0.770, 1.15, and 1.54 radians/second. During the formation flying task, the relative positions of the lead aircraft and the simulator were recorded, as were the pilot control inputs. At the end of each trial, the pilot subjects were asked to give a rating of the aircraft handling qualities, as presented in the simulator, according to the Cooper-Harper rating scale.[18] After four trials of the formation flight, the simulator was re-initialized at an altitude of 10 meters above a simulated canyon scene. The pilots then performed a series of dash-quick-stop maneuvers and sinusoidal pitch oscillation maneuvers. At the end of these flight tasks, they were asked specifically to rate the motion of the simulator using a seven component rating system designed for this purpose. The motion rating scale used numerical ratings of smoothness, sense, amplitude, phase lag, discomfort, and disorientation as well as an overall rating of the motion relative to fixed base operation.

Results of VTOL Simulation

The instructions given to the pilot subjects were to maintain relative position during the formation flight. However, due to the difficulty of judging distance, given the limitations of the visual scene and the 33m separation between the lead aircraft and the simulator, the relative velocity between the two aircraft proved to be a more appropriate measure of performance. The velocity difference between the lead aircraft and the simulator was computed as a velocity error. The performance of each pilot was scored by computing the variance of the velocity error and recording this as a Velocity Error Score (VES). The variance of the velocity error was used for the VES instead of the root-mean-square velocity error to eliminate the effect of a steady state velocity error. For this reason, the VES is a more appropriate measure of the correlation of the velocity of the simulated VTOL aircraft and the target aircraft. Despite the fact that the pilots were highly trained in VTOL aircraft, their performance varied widely, eliminating the possibility of combining measurements across subjects. For illustration in this discussion, data taken from a single pilot (Pilot #3) is presented. The observations and conclusions drawn from this data are generally applicable to all four pilot subjects.

The velocity error scores for Pilot #3 given in Figure 5 indicate that the greatest differences in tracking performance occur between fixed base and any of the motion conditions. The differences in tracking performance when motion cues were present were relatively small. The fact that pilot performance appears to be robust in the presence of significantly different motion conditions is a limitation inherent in the use of performance alone as an indicator of motion fidelity, particularly when highly skilled pilots are used as test subjects.

In order to examine the effects of motion conditions on pilot control behavior, a pilot describing function analysis was performed. Using the disturbance input to the lead aircraft, the pilot cyclic pitch inputs, and the model aircraft dynamics, the linear portion of the pilot control response was reconstructed. The data indicated a large nonlinear component (remnant) in all cases and there were no reliable differences in the pilot describing function among the motion conditions, including fixed base.

The level of pilot compensation required to perform the formation flying task for each of the motion conditions is reflected in the Cooper-Harper ratings presented in Figure 6 (a high Cooper-Harper Rating reflects poor handling qualities). The relatively high ratings (4 to 6) given for the handling qualities of the simulator in the formation flying task indicate that considerable pilot compensation was required to achieve adequate task performance. Differences in required pilot compensation among motion conditions appear to be more significant than differences in task performance.

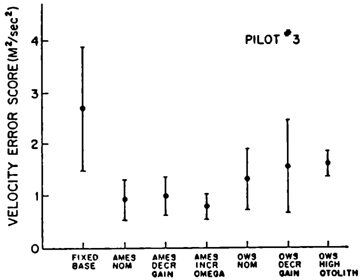


Figure 5. Pilot performance in longitudinal tracking task (formation flight) versus motion condition.

The slight correlation (R = 0.61) between performance and the Cooper-Harper rating assigned indicates that pilot compensation may be a more sensitive measure of the role of motion cues in the formation flying task.

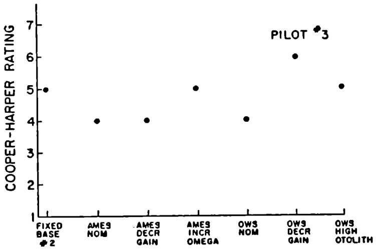


Figure 6. Pilot ratings of vehicle handling qualities versus motion condition during longitudinal tracking task (formation flight).

The Ames Nominal, Ames Decreased Gain, and OWS Nominal received equal ratings (CH = 4), indicating that the simulator required less pilot compensation to fly with these washout systems than it did under Fixed Base (CH = 5). The Ames Increased Omega and OWS High Otolith Weighting (CH = 5) were judged equivalent to Fixed Base and the OWS Decreased Gain (CH = 6) was judged poorer than Fixed Base.

In general, there was some correlation (R = 0.72 to 0.91) for all pilots between the Cooper-harper handling quality rating assigned to the formation flying task under each motion condition and the overall rating of simulator motion. Each pilot, however, exhibited a different correlation between components of the motion rating scale and the overall motion rating, indicating considerable between-subject variation in the assessment of each component of simulator motion. This difference among individuals makes comparison of ratings across a pool of pilot subjects quite difficult. A similar effect has been observed for the subjective rating of mental workload and techniques exist to reduce the between-subject variation in workload ratings.[19] These same techniques could be applied to the motion rating scales described here.

Evaluation of Motion Platform Alternatives

Transport Category Aircraft Simulation

Assessing motion platform effects on pilot performance in tasks representative of those which are required during transport category aircraft training and checking operations is a first step in identifying candidate simulator motion systems which could be evaluated in a training environment. We chose to apply the vestibular model to compute vestibular error and use it as a quantitative index of simulator performance.

Eighteen air transport pilots, currently flying Boeing 727 aircraft, participated as paid volunteers in the study. Of the eighteen pilots, three served in the Captain and fifteen in the First Officer crewmember position. Experience in the 727 ranged from 3 months to 7.5 years with an average of 2.4 years. A Boeing 727-200 flight simulator certificated under Phase II of the Federal Aviation Regulations simulator requirements section (Part 121, Appendix H) was used for the study. The simulator, which is located at NASA Ames Research Center provides a full six DOF motion utilizing a nonlinear, adaptive motion drive logic scheme. A dusk/night visual system provides a computer generated image of the out-of-the-cockpit scene to both the Captain and First Officer positions. For this study, only night scenes were presented.

Three motion platform conditions were compared in this study: the full six DOF motion required for Phase II simulators and two limited motion conditions. The latter platform motion conditions were provided by restricting the software logic driving the platform. For one of the limited motion conditions, the six DOF system was reduced to two DOF: vertical and lateral translational motion. Amplitude of normal platform motion excursion in these two axes was not limited. In the second limited motion condition, small amplitude vertical translation motion commonly called "special effects" were the only motion cues provided. These special effects included the following: runway touchdown bump, vibrations induced by runway roughness, buffets associated with flap, landing gear, and spoiler extension, and Mach and stall buffet. Maximum leg extension with these effects was .63 cm. These special effects were provided in the six DOF and two DOF motion conditions as well.

Experimental Design

Six of the eighteen pilots were randomly assigned to each of three test scenarios. The three test scenarios were constructed to allow the evaluation of pilot performance in task conditions representative of those they would receive in the operational training environment. An additional criterion for task selection was the desire that significant pilot control activity be involved. This criterion was included to increase the probability of detecting motion platform effects if they did, in fact, exist. Each pilot was tested individually with the pilot-not-flying duties performed by a research pilot. The three test scenarios were as follows: (1) engine flameout on takeoff subsequent to rotation; (2) an airwork scenario consisting of steep turns, approach to stall, and standard rate turns at altitude with yaw dampers failed; and (3) an ILS approach and landing flown through a low-level, horizontal windshear. All scenarios were conducted in and around the simulated San Francisco International Airport environment. With the exception of the ILS approach and landing, all maneuvers were conducted in standard day, no wind, visual meteorological conditions. The simulated aircraft had a takeoff weight of 67,300 kg. In order to standardize testing, fuel quantities were held constant throughout the flights.

Prior to testing, pilots were provided with the opportunity to fly VFR approaches and landings with full platform motion in order to become familiar with the simulation environment. Pilots were not informed that motion platform conditions would be altered, only that the study's intent was to assess simulator fidelity issues. In all motion test conditions, all normal procedures involving full motion operations were conducted so that pilots would not be made aware of any changes in platform functioning prior to testing. The order in which the three motion conditions were tested was counterbalanced across the six pilots who flew the scenario.

Both subjective pilot ratings and objective simulator measurements were taken during the course of the study. The pilot ratings were taken after the completion of testing on a given motion condition within each scenario. The rating instrument consisted of six items, each requiring a response on a 5-point scale. A rating of 3 on this scale indicated that the pilot felt the simulator to be very similar to the aircraft. For example, a rating of 1 on control workload was given if the simulator control effort was much less than that of the aircraft, a 5 if the effort was much more than that of the aircraft. The six items addressed the following: total control workload in the scenario, control workload during configuration changes, general responsiveness of the simulator to control inputs, the utility of the simulator for training and checking, and an assessment of overall realism of the simulation. For all items, pilots were asked to base their ratings to the extent possible on experience with the aircraft. Objective measures of pilot and simulator performance were collected in real time at a rate of 15 samples per second. Aircraft state parameters such as airspeed, attitude, and altitude were sampled as were measures of simulator motion, the output of the spatial orientation models, and pilot control inputs.

Pilot Rating of Simulator Fidelity

The pilots' subjective ratings of the simulator fidelity are depicted graphically in Figure 7. This figure shows the rating for each of the six categories averaged across the eighteen pilots and three test scenarios. In all categories and in all motion platform conditions, the pilots rated the simulation to be very similar to the aircraft. No reliable differences in pilot ratings were found for the three motion conditions, either within or across test scenarios for the six rating categories.

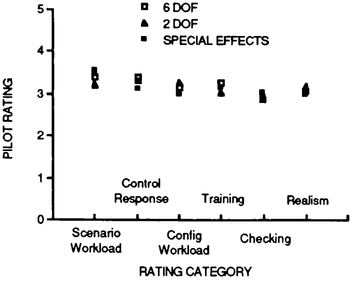


Figure 7. Pilot rating of simulator fidelity as a function of motion condition.

Pilot Performance Results

Aircraft state parameters and pilot control activity were analyzed to determine the effects of platform motion variations on pilot performance. Where successive trials of the same maneuver were executed, data from these trials were averaged. Statistical analyses for a repeated measures design were conducted to determine whether differences among platform motion conditions were reliable.

For all three scenarios, no statistically reliable differences in pilot performance among the three motion conditions were observed. Figure 8 illustrates a typical comparison of pilot performance for the engine flameout scenario for each of the motion conditions. Differences in other performance measures were similar to and typical of the other two scenario types.

Spatial Orientation Error

The dynamic models of the vestibular system described above were used to estimate the mean root-mean-square (RMS) vestibular error. Typical results of these computations are depicted in Figures 9 and 10 for the instrument landing scenario and are typical of all scenarios. In general, the vestibular error in roll, pitch, and yaw was sub-threshold for all three motion conditions. Vestibular error computed in the vertical axis (heave) was several times the threshold value but was sub-threshold in the longitudinal axis (surge). In general, no reliable differences in mean RMS vestibular error were found among motion conditions.

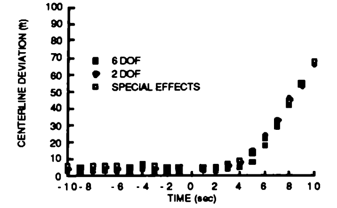


Figure 8. Aircraft runway centerline deviation prior to and following engine flameout (at time = 0) as a function of motion condition.

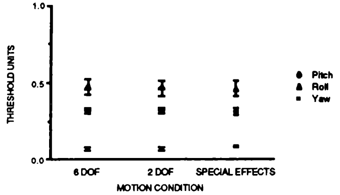


Figure 9. Mean RMS vestibular error in pitch, roll, and yaw computed for the instrument approach scenario.

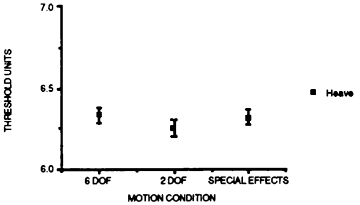


Figure 10. Mean RMS vestibular error in the vertical axis (heave) for the instrument approach scenario.

Discussion

The absence of any reliable effects of platform motion on either subjective assessments of simulator fidelity or performance of the pilots is of particular interest since the absolute accelerations provided by the full motion conditions were, in general, well above threshold for the perception of whole body motion. However, when the vestibular model is used to estimate the difference between sensed motion in the simulator and the actual aircraft, the differences between full six DOF simulator platform motion and limited special effects motion alone are not significant. Whether whole-body motion affects pilot behavior is dependent on a number of factors. The more important of these are the dynamics of the vehicle being simulated, the nature and extent of pilot experience in aircraft and simulator operations, the task environment including the availability of information redundant to that provided by motion alone (e.g. a wide field-of-view visual scene), and pilot attitudes and beliefs. In the VTOL study described above, the pilots were aware that platform motion was being manipulated and were sensitive to differences in motion condition. In the transport aircraft study, the pilots were unaware that motion was being manipulated and this, combined with the overall realism of the simulation and the dynamics of the aircraft (a transport category fixed-wing aircraft as opposed to a hovering vehicle) may have overwhelmed the small differences in motion sensed as computed by the vestibular model.

Conclusion

A model for the perception of human spatial orientation based on physiological models of the vestibular organs has been successfully applied to the problem of flight simulator motion fidelity design and evaluation. The optimal control approach to the design of platform motion controllers is capable of producing motion drive logic that has been demonstrated to be comparable to an existing empirically optimized washout system in terms of pilot performance and simulator acceptability. The advantage of the model-based optimal design technique in that it permits the manipulation of the motion controller performance not only in terms of the motion platform displacement and accelerations, but directly at the level of the pilot's modeled motion perception. The motion controller synthesized with the optimization technique may be generated off-line by assuming that the pilot inputs and aircraft disturbances are a stochastic process. It is theoretically possible to improve the performance of an optimal washout system by measuring pilot inputs during the simulation and optimizing platform motion on-line, given the instantaneous state of the motion system; however, the computation time required to perform this is currently prohibitive. The emerging technology of high-speed parallel processors may provide a solution to this problem.

While the transport category aircraft simulation study did not address the issue of whether motion platform cues affect pilot behavior under all possible conditions of flight, from the standpoint of normal trainingand checking operations, the subjective and objective data collected suggest that large, complex motion platform systems may not be necessary for either reasons of pilot acceptance or performance, given the presence of a wide field-of-view visual scene and sufficient "special effects" motion to enhance the realism of the simulation. For this type of aircraft (Boeing 727), simulators with very limited motion capability may be adequate for training purposes. Further research on this issue is warranted. Caution should be exercised in any attempt to generalize from these data to other transport aircraft simulations. For example, substantial asymmetric thrust effects in aircraft with wing-mounted engines may produce lateral accelerations that differ markedly from those produced in the 727. Finally, the study evaluated the behavior of experienced 727 pilots. It remains to be determined whether motion plays a significant role in the acquisition of flying skills in the simulator and if the transfer of these skills to the aircraft is affected by the absence of large amplitude platform motion.

Finally, the effect of wide field-of-view visual scenes upon the pilot must be incorporated in future models of spatial orientation perception used in flight simulation. As new models are created for the interaction between visual and vestibular cues in the human pilot's perception of spatial orientation, they may be used in a manner similar to that described above as engineering tools for the flight simulator designer.

Acknowledgement

We would like to extend our appreciation to John Stewart and Richard Bray of NASA Ames Research Center for their assistance during the design and performance of the VMS experiments. This research was performed with the support of the National Aeronautics and Space Administration under grant NAG 2-12.

References

1. Waag, W.L., "Training effectiveness of visual and motion simulation", AFHRL-TR-97-72, Williams AFB, Arizona: Oper. Train, Div., USAF Human Res. Lab., 1981.

2. Young, L.R., and Meiry, J.L., "A revised dynamic otolith model", Aerospace Medicine, 39(16): 606-608, June, 1968.

3. Young, L.R., and Oman, C.M., "Model for vestibular adaptation to horizontal rotation", Aerospace Medicine, 40(10): 1076-1080, October, 1969.

4. Ormsby, C.C., and Young, L.R., "{Perception of static orientation in a constant gravito-inertial environment", Aviation, Space, and Environmental Medicine 47(2): 159-164, February, 1976.

5. Shirley, R.S., and Young, L.R., "Motion cues in man vehicle control", IEE Transactions on Man Machine Systems, MMS-9(4): 121-128, December 1968.

6. Curry, R.E., Hoffman, W.C., and Young, L.R., "Pilot modeling for manned simulation", AFFDL-TR-76-124, Air Force Flight Dynamics Laboratory Publication, Vol. 1 December, 1976.

7. Zacharias, G., and Young, L.R., "Influence of combined visual and vestibular cues on human perception and control of horizontal rotation", Experimental Brain Reserach, 41: 172-183, 1981.

8. Steinhausen, W., "Uber den Nachweis der Bewegung der Cupula in der intakten Bogengagsampulle des Labyrinthes dei der naturlichen rotatorischen und kalorischen Reizung", Pflugers Arch. ges. Physiol., 228: 322-328, 1931.

9. van Egmond, A.A.J., Groen, J.J., and Jongkees, L.B.WW., "The mechanics of the semicircular canal", Journal of Physiology, 110:1-17, 1949.

10. Young, L.R., "Role of the vestibular system in posture and movement", in Medical Physiology, 13th Edition, V.B. Mountcastle (ed.), C.V. Mosby Co., St. Louis, MO, Vol. 1:704-721, 1974.

11. Fernandez, C., and Goldberg, J.M., "Physiology of peripheral neurons innervating semicircular canals of the squirrel monkey, II: Response to sinusoidal stimulation and dynamics of peripheral vestibular system", Journal of Neurophysiology 34(4): 661-675, 1971.

12. Fernandez, C., and Goldberg, J.M., "Physiology of peripheral neurons innervating otolith organs of the squirrel monkey, II: Directional selectivity and force-response relations", Journal of Neurophysiology, 39(5): 985-995, 1976.

13. Hosman, R.J.A.W., and van der Vaart, J.C., "Vestibular models and thresholds of motion perception. Results of tests in a flight simulator", Report LR-265 Department of Aerospace Engineering, Delft University, Delft - The Netherlands, April 1978.

14. Sivan, R., Ish-Shalom, J., and Huang, J.K., "An optimal control approach to the design of moving flight simulators", IEEE Transactions on Systems, Man, and Cybernetics, SMC-12(6): 818-827, November/December, 1982.

15. Zarchan, P., "Representation of realistic evasive maneuvers by the use of shaping filters", Journal of Guidance and Control, 2(4): 290-295, July 10979.ATH

16. Kwakernaak, H., and Sivan, R., Linear Optimal Control Systems, Wiley-Interscience, New York, 1972.

17. Sinacori, J.B., "A brief survey of motion simulators' drive logic with emphasis on the roll axis", Systems Technology, Inc. WP-1094-2, May 1977.

18. Cooper, G.E., and Harper, R.P., Jr., "The use of pilot rating in the evaluation of aircraft handling qualities", NASA Tech. Note, NASA TN D-5153, April, 1969.

19. Hart, S.G., and Staveland, L.E., "Development of a multi-dimensional workload rating scale: Results of empirical and theoretical research", Human Mental Workload, North Holland Press, Amsterdam, 1986.

THE DYNAMIC SEAT AS AN ANGULAR CUING DEVICE: CONTROL OF ROLL AND PITCH VS. THE CONTROL OF ALTITUDE AND HEADING

Jeffrey D. Cress
Systems Research Laboratories, Inc.
Dayton, OH 45440

Grant R. McMillan
Armstrong Aerospace Medical Research Laboratory
Wright-Patterson Air Force Base, OH 45433

Michael J. Gilkey
Systems Research Laboratories, Inc.
Dayton, OH 45440

ABSTRACT

A series of experiments is currently being conducted to examine the effectiveness of the dynamic seat as a motion display for multi-degree-of-freedom flight simulation tasks. In previous studies, the dynamic seat has significantly reduced tracking error for roll and pitch control tasks. However, attempts to utilize this device for linear cuing have met with less success. In this study, six subjects performed two sets of tasks: 1) roll and pitch control, and 2) heading and altitude control; roll and pitch seat motion cues were provided for both sets of tasks. Subjects were required to maintain the pre-specified variables in the presence of pseudo-random roll-rate and pitch-rate disturbances. Subjects also performed these tasks with no seat motion to establish a baseline performance level. Examination of the asymptotic data revealed that the dynamic seat reduced tracking error by slightly greater than 30 percent for roll, pitch and heading control, while reducing altitude error by only 8 percent. Subjects also completed several trials at varying seat gains, providing data for an investigation of the relationship between performance and the amount of seat motion. It was discovered that seat motion can be reduced by 50 percent without significantly affecting performance; however, performance falls off sharply at lower gains.

INTRODUCTION

The Human Engineering Division at Wright-Patterson Air Force Base has been studying the effectiveness of the dynamic seat as a motion cuing device since 1980. A series of experiments is currently being conducted to establish effective drive laws for multi-degree-of-freedom angular and linear motion cuing. Our previous research in the area of roll-axis cuing showed that driving the seat proportional to roll angle, roll rate [1] or their combination was most effective. [2] These experiments have demonstrated that the dynamic seat can effectively provide onset motion information for roll-axis tasks. Knowledge gained from the roll-axis research was applied to the pitch-axis, resulting in similar performance benefits, [3] however, our recent attempts to utilize this device for linear cuing have been somewhat less successful.

For the first linear-cuing experiment, an altitude regulation task was employed with the seat's vertical position being proportional to either the aircraft's normal acceleration or normal velocity. Subjects' altitude error was only slightly smaller with either of the two algorithms than it was with no seat motion. While acceleration normal to the aircraft is a realistic cue that would seem to be useful for the control of altitude, it appeared to have little effect on the altitude tracking error. It was hypothesized that the seat's lack of effect on performance was due to the indirect relationship between that which was being scored (altitude) and that which was being displayed by the seat (aircraft's normal acceleration). To further explore this hypothesis a second altitude regulation study was performed where the seat's vertical position was driven with either altitude, altitude rate or altitude acceleration. Examination of the results revealed that the seat now significantly reduced tracking error. Unfortunately, the seat motion was not realistic since aircraft do not provide motion cues directly proportional to altitude or its derivatives. Nevertheless, the results suggest that a strong scored variable / seat motion relationship is crucial for effective motion cuing. This paper describes further research conducted to resolve this issue.

For this experiment we concentrated on two sets of disturbance regulation tasks: one for which the relationship between seat motion and the scored variable was direct and one for which the relationship was not as strong. Roll and pitch control was chosen for the first set of tasks; heading and altitude control for the second, with the seat displaying roll and pitch for both sets of tasks. Two follow-on studies were also conducted examining the effect of the magnitude of seat motion on performance of the aforementioned tasks.

METHODS

Subjects. Six subjects with normal or corrected-to-normal vision participated. None of the subjects were pilots.

Apparatus

Simulated Aircraft. A fighter type aircraft was simulated on a Digital PDP 11/60 computer using simplified dynamics (Figure 1). The simulated aircraft was controlled using a side-mounted force sensitive (isometric) control stick. Aircraft roll was regulated through lateral inputs while fore-aft inputs affected pitch. Roll and pitch gains were 112.0 and 10.9 degrees/second per newton meter of torque, respectively, with a breakout force of 2.2 newtons for both axes.

Disturbances. The simulated wind gusts were implemented using spectral shaping filters, based on the Dryden gust model, driven by a sum-of-sines approximation to white noise. The disturbance for each axis consisted of ten harmonically unrelated sinusoids, ranging from 0.049 to 6.611 Hertz for the roll-rate disturbance and 0.029 to 5.403 Hertz for the pitch-rate disturbance, with the two spectra interleaved. The starting phase for each sinusoid was randomly selected before each trial to eliminate the learning of disturbance patterns.

Display. The roll, pitch, heading and altitude of the simulated aircraft were determined by the sum of the control inputs from the subject and the simulated wind gusts. The resulting states were then sent to a high resolution raster-graphics system (Silicon Graphics IRIS) and depicted on a color monitor. The monitor measured .29 meters high by .39 meters wide and was located approximately .74 meters from the subject, resulting in a 22 degree by 30 degree field of view. The visual scene consisted of a green grid on a black background depicting a perspective view of flat terrain as seen from the aircraft traveling at a speed of 128 meters/second (Figure 2).

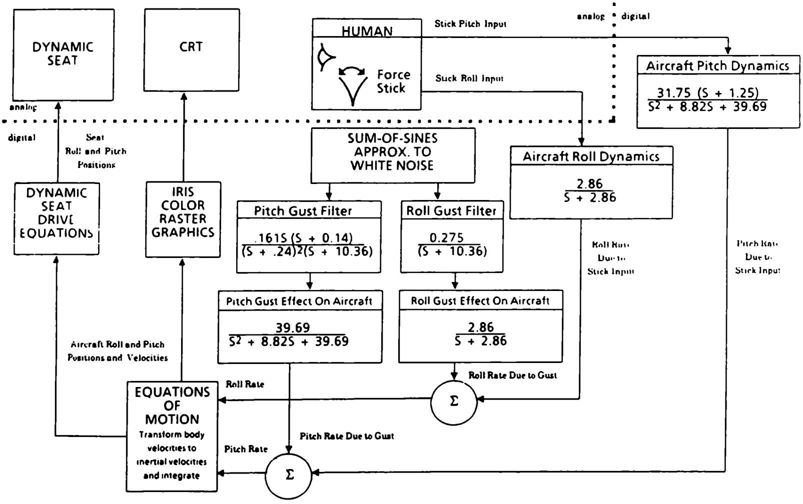
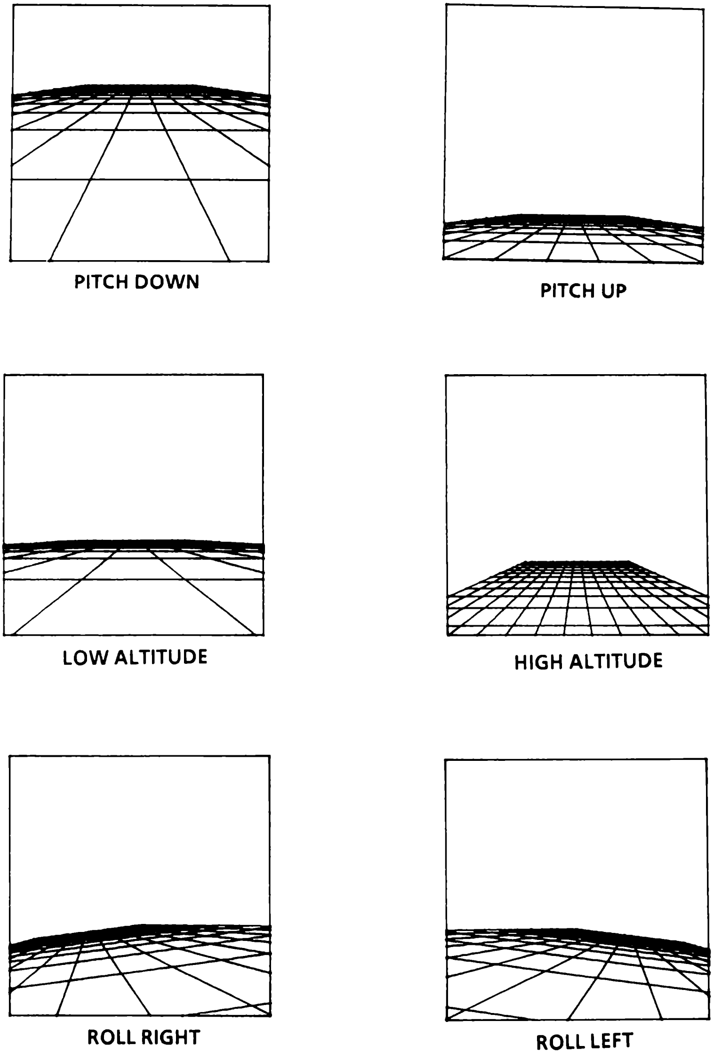

DYNAMIC SEAT
CRT
HUMAN
Force Stick
Stick Pitch Input
analog
digital
Aircraft Pitch Dynamics
$\frac{31.75\,(S + 1.25)}{S^2 + 8.82S + 39.69}$
Stick Roll Input
analog
digital
Seat Roll and Pitch Positions
SUM-OF-SINES APPROX. TO WHITE NOISE
Aircraft Roll Dynamics
$\frac{2.86}{S + 2.86}$
DYNAMIC SEAT DRIVE EQUATIONS
IRIS COLOR RASTER GRAPHICS
Pitch Gust Filter
$\frac{.161S\,(S + 0.14)}{(S + .24)^2(S + 10.36)}$
Roll Gust Filter
$\frac{0.275}{(S + 10.36)}$
Roll Rate Due to Stick Input
Pitch Rate Due to Stick Input
Pitch Gust Effect On Aircraft
$\frac{39.69}{S^2 + 8.82S + 39.69}$
Roll Gust Effect On Aircraft
$\frac{2.86}{S + 2.86}$
Aircraft Roll and Pitch Positions and Velocities
Roll Rate Due to Gust
EQUATIONS OF MOTION
Transform body velocities to inertial velocities and integrate
Roll Rate
Σ
Pitch Rate Due to Gust
Pitch Rate
Σ

Figure 1. System Block Diagram



PITCH DOWN

PITCH UP

LOW ALTITUDE

HIGH ALTITUDE

ROLL RIGHT

ROLL LEFT

Figure 2. Display.

Dynamic Seat. The Advanced Low-Cost G-Cuing System (ALCOGS) is a device that displays both onset and sustained motion information (Figure 3).[4] The seat pan is supported by hydraulic actuators, one at each of the back two corners and one in the center at the front of the seat. The rear actuators are used to roll the seat about the longitudinal axis while the front seat pan actuator pitches the seat about its rear edge. There are also three actuators for the backrest, one located in the center at the top and the other two located at the bottom corners. The top backrest actuator (the only one used for this study) pitches the backrest about its bottom edge. The seat pan and backrest pitched as if they were a single unit.

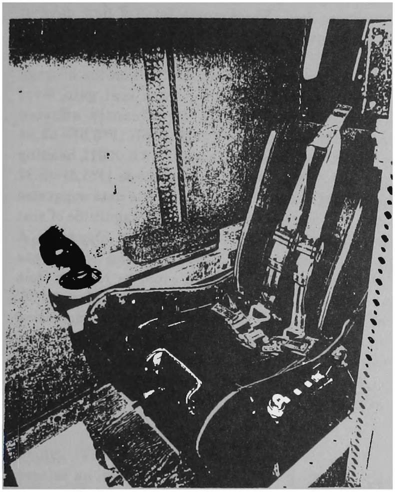


Figure 3. Dynamic Seat

Seat Motion. Roll and pitch seat motion was used for both sets of tasks. The drive algorithm for the seat motion was based on the zero and first derivatives of roll (pitch).[2] The actual equations are as follows:

$$\text{seat roll (pitch) angle} = K1 \times [\text{ A/C roll (pitch) angle} + K2 \times \text{A/C roll (pitch) rate }]$$

where: $K2 = 0.167$ and
$K1 = 0.67, 0.335, 0.1675, 0.0833, 0.0616$ or 0.0

The six different values for K1 represent the six different levels of seat motion used in the second phase of the experiment (K1 = 0.0 was used for the static condition). For the first phase of data collection, K1 = 0.335 was used for the motion condition.

Task Descriptions. For the roll and pitch regulation tasks, subjects were instructed to maintain zero degree roll and pitch attitudes while errors were induced by the sum-of-sines disturbances. Deviations in heading and altitude were neither displayed nor scored.

For the heading and altitude tasks, subjects were told to fly parallel with the longitudinal grid lines and to maintain an altitude of 30.5 meters in the presence of the sum-of-sines disturbances. Roll and pitch deviations were visually displayed since they are necessary information for the control of heading and altitude, however, they were not scored.

ROLL AND PITCH VS. HEADING AND ALTITUDE CONTROL STUDY

Procedures

Experimental Design. A completely within subjects design was used; all subjects performed both the roll/pitch and the heading/altitude disturbance regulation tasks with and without seat motion.

Familiarization, Training and Data Collection. During the familiarization phase, each subject was allowed several minutes of unconstrained flight (roll and pitch tasks with no seat motion) while being verbally coached. This was done to acquaint the subjects with control/display relationships as well as control stick sensitivities. Subjects then ran two static trials in which they attempted to maintain zero degree roll and pitch attitudes in the presence of pseudo-random roll-rate and pitch-rate disturbances as described earlier. Subjects were then given two more trials with seat motion added. Each step was then repeated for the heading and altitude maintenance tasks. At the end of each trial, subjects were presented the mean, standard deviation and root-mean-squared values for each of the criterion variables. Subjects participated in a total of ten days of experimentation. Each day, subjects experienced four, 102 second, trials for each of the following conditions: 1) roll pitch - motion, 2) roll/pitch - static, 3) heading/altitude - motion and 4) heading/altitude - static. The order in which the subjects received the conditions was random.

Results. Learning curves were plotted for each subject and variable to ensure that asymptotic performance was reached. Plotting the mean scores verses trial revealed that the subjects' ability to reduce mean error (constant error / bias) was a function of the task type. For the roll and pitch tasks, subjects' mean scores approached zero; however, mean scores for the heading and altitude tasks did not appear to improve as much. This suggests that there were differences in the quality of the visual references among the individual tasks. To eliminate the differences in visual reference as a contribution to tracking error, error standard deviation was chosen as the measurement of performance.

The means of the error standard deviations for the last 8 trials were computed for roll, pitch, heading and altitude under motion and static conditions. The percent reduction in tracking error produced by the dynamic seat was then calculated using the following formula:

$$PERCENT\ REDUCTION = \frac{SD\ static - SD\ motion}{SD\ static}$$

An analysis of variance was performed on these values with task type specified as an independent variable. The ANOVA revealed that the values were significantly affected by task type [F(3,15)=8.86, p<0.0013]. A Tukey's comparison of means showed that dynamic seat motion had a significantly larger effect on roll, pitch and heading control than on altitude control (see Figure 4).

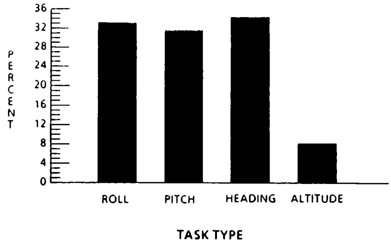


Figure 4. Group averages of the percent reduction in tracking error produced by the dynamic seat.

SEAT GAIN PHASE 1

Procedures

Experimental Design and Data Collection. After the previous ten days of experimentation, subjects were brought back for further test runs. For this phase of the experiment, subjects performed both sets of tasks with six different levels of seat motion in order to determine the effect of seat motion magnitude on tracking performance. Each day, subjects received the highest seat gain value first, progressing through the lowest seat gain to the static condition. Subjects were then given the seat gains in reverse order. Only one set of tasks was presented in a day, resulting in two trials for each of the gain levels per day. There was a total of four days of experimentation, two days for each set of tasks.

Results. The results of an ANOVA on the average error standard deviation at each seat gain level indicated that seat gain significantly affected performance for all four tasks: roll [F(5,5)=52.64 p<0.0001], pitch [F(5,5)=44.81 p<0.0001], heading [F(5,5)=8.86 p<0.0001] and altitude [F(5,5)=3.32 p<0.0194]. Visual inspection of the data suggested that performance improved as the magnitude of seat motion increased for each of the tasks (Figure 5). A Tukey's comparison of the roll and pitch data indicated that the seat gain could be reduced from the level used in the first experiment (K1 = 0.335) by a factor of two without significantly increasing tracking error. Heading error was significantly affected only if seat gain was reduced by a factor of eight while altitude error was affected only if no seat motion was present.

SEAT GAIN PHASE 2

Procedures

Experimental Design and Data Collection. Four subjects from the first seat gain experiment were brought back to determine if subjects could learn to perform well with the lower seat gains if given additional training. For this phase, the subjects completed 64 trials of the roll and pitch tasks, the first 56 of which used a seat gain value 1/4 of that used in the initial study (K1=0.0833). The final 8 trials used the original (K1=0.335) seat gain. The 0.0833 seat gain was chosen because it was the largest seat gain that yielded performance scores which were significantly worse than those yielded by the 0.335 seat gain. Our hypothesis was that, given additional practice with the 0.0833 seat gain, there

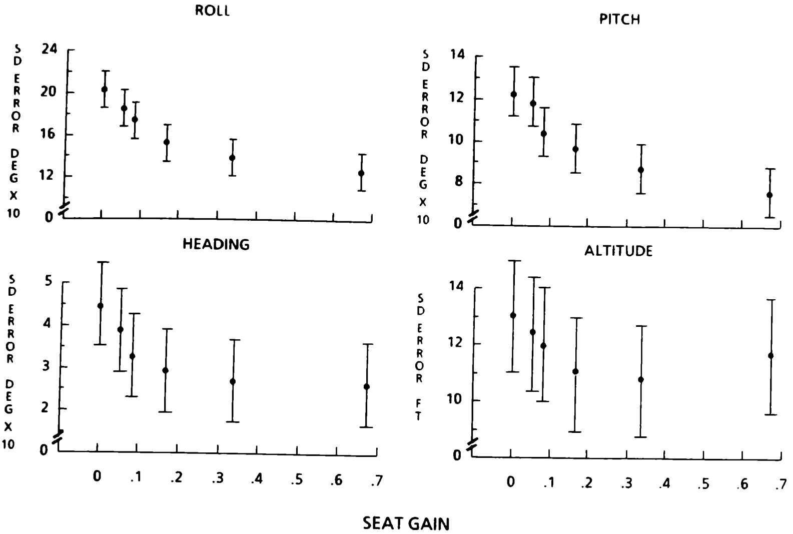


Figure 5. Tukey's 95% Confidence Intervals of group performance at the different levels of seat motion. The 0.335 gain was used in the previous study and was the reference value used for statistical comparisons.

would be no significant difference in performance with either of the seat gains. The roll and pitch maintenance tasks were chosen because they were the most sensitive to changes in seat gain.

Results. Visual inspection of the learning curves revealed an immediate drop in tracking error when seat gain was increased at trial 57. An ANOVA comparing the average of both the last eight 0.0833 trials and the eight 0.335 trials showed the drop in tracking error to be statistically significant: roll [F(1,3)=48.66, $p<0.006$] and pitch [F(1,3)=19.60 $p<0.0214$]. Thus, the results did not support our hypothesis that, given additional training, there would be no significant difference in performance with the two seat gains.

DISCUSSION

The effects of seat motion on tracking performance for the roll and pitch tasks were consistent with findings from previous experiments performed in this lab. The same can be said for the small reduction in altitude error produced by pitch cuing. The contribution of roll motion to the control of heading, however, was expected to be much smaller, closer to the results from the altitude maintenance task. This surprising result may be due in-part to the aircraft dynamics. By comparing the roll-to-heading and pitch-to-altitude relationships, it becomes apparent that the two relationships are not analogous. First, roll and pitch are angular degrees-of-freedom as is heading; altitude, on the other hand, is a linear degree-of-freedom. Second, heading depends primarily on the aircraft roll and pitch states (especially roll). While altitude is affected by roll and pitch, it is also heavily dependent on angle of attack (not presented by the seat). This missing piece of motion information may be partially responsible for the seat's lack of effect. Further research will be necessary to answer this question. However, viewing the results from this perspective, our initial theory (a strong scored variable seat motion relationship is necessary for effective motion cuing) is supported.

Attempts to determine the effect of seat motion magnitude on performance were a result of pilot complaints that "the seat's lack of realism was due to too much motion". However, previous research has shown that performance improves with higher seat gains and the results from this experiment support those findings. The initial hypothesis was that subjects were proficient with the higher seat gain because of extensive practice and weren't given sufficient time to become proficient with the lower gains. This hypothesis was dismissed following the second seat gain study. Even after extensive training with the low gain, subjects still performed significantly better with the higher gain. A more probable hypothesis is that at asymptotic performance, the roll and pitch errors are small enough that seat motion is approaching threshold at the lower seat gains. While there may be a point at which the seat gain is too high to be beneficial, the values chosen for this study are apparently not large enough to show this.

The lack of effect of seat gain on altitude error was expected since seat motion did not prove to be very useful for the control of altitude, but it is not clear why performance on the heading task was not more susceptible to changes in seat gain. This problem may also be solved with further research.

REFERENCES

1. Martin, E.A. (1985). The influence of tactual seat motion cues in training and performance in a roll-axis compensatory tracking task setting. Unpublished doctorial dissertation, The Ohio State University, Columbus, OH.

2. Flach, J.M., Riccio, G. E., McMillan, G.R. and Warren, R. (1986). Psychophysical methods for equating performance between alternative motion simulators. Ergonomics, 29. Columbus, OH.

3. Osgood, R.K., Taylor, K., and McClurg, T. (1988). The dynamic seat as an angular motion cuing device. Presented at the Human Factors 28th Annual Meeting. Denver, CO.

4. Kleinwaks, J.M. (1980). Advanced low cost G-cuing system (ALCOGS) (Technical Report AFHRL-TR-79-62). Brooks AFB, TX.

5. Martin, E. A., Osgood, R. K., and McMillan, G.R. (1987). The dynamic seat as an onset cuing device. Presented at AIAA Flight Simulation Technologies Conference,. Monteray, CA

6. Puig, J.A., Harris, W.T., and Ricard, G.L. (1978) Motion in flight simulation: An annoted bibliography. Technical Report NAVTRAEQUIPCEN IH-298, NTSC, Orlando, FL

ACKNOWLEDGEMENTS

The authors would like to thank Matt Middendorf for his direction and support during this project.

89-3337-CP

G-SEAT HEAVE MOTION CUEING FOR IMPROVED HANDLING IN HELICOPTER SIMULATORS

A.D. White

Flight Management Department FM3 Division
Procurement Executive, Ministry of Defence
Royal Aerospace Establishment, Bedford, UK

Abstract

Although g-seats have been used for many years to produce sustained 'g' cues on fast-jet simulators, little attention has been given to their use in providing motion onset cues, especially the lower level 'g' onset cues typical of helicopter flight. The aim of this investigation was to determine whether a hydraulic g-seat can provide beneficial cueing in a Nap of the Earth (NOE) environment. The g-seat has been evaluated using an agile helicopter model and tasks which emphasise manoeuvring in the vertical plane. Results indicate that the seat not only greatly enhances the sense of realism of the simulator, but also enables pilots to control vehicle height more realistically and effectively. Analysis of the data supports pilots' subjective impressions that performance was improved when the g-seat was in operation, typically showing reduced overcontrol and more consistent handling, even in hover.

Introduction

The simulation of motion in the heave axis (the aircraft's normal acceleration axis) has always been a problem for flight simulation engineers[1]. Motion platforms can provide initial transient (or onset) cues, but accelerations cannot be sustained for long if the platform is to remain within its excursion limits. In addition, unlike sustained sway and surge motion, which can be simulated by rotating the cockpit in roll and pitch, heave motion can only be simulated realistically by moving the cockpit physically up and down. This means that the high levels of normal acceleration (or 'g') experienced in aircraft manoeuvres cannot be reproduced. Over the last two decades attempts have been made to improve the sensation of heave motion either by increasing the range of motion platform displacement (as in the VMS at NASA Ames), which is intended mainly to stimulate the vestibular senses, or by using devices such as g-seats, g-suits and helmet loaders to stimulate the somatic (body feel) and kinaesthetic (limb orientation) senses directly.

G-seats were originally conceived as a means of improving the perception of sustained acceleration, the idea being that the seat would sustain the impression of motion after the onset cue from platform motion had been washed out. Since they were designed to give sustained cues, the bandwidth requirement for these seats was low and they were usually quite slow to follow commands. Recent devices have the response necessary to generate motion onset cues.

A number of factors suggested that a high-bandwidth g-seat might prove to be well suited to helicopter simulation.

- o Initial experience with a g-seat at RAE Bedford had shown that, although useful cueing is achieved up to high 'g' levels, the cues resemble aircraft cues only at low levels.
- o Vibration cueing supplied through the seat had already been well received by pilots. The ability to provide both vibration and heave motion cues in a single device could prove attractive for lower-fidelity training devices with no other motion cues or to complement platform motion.
- o Helicopter operations are becoming increasingly demanding, as indicated by the current trend towards improved agility and performance. Agility (ie the ability to achieve a manoeuvre quickly) requires that simulator motion devices have a high bandwidth to enable the necessary onset cues to be perceived correctly.
- o Unlike conventional fixed-wing aircraft, helicopters are capable of performing large vertical manoeuvres without first having to rotate, and such manoeuvres require good vertical motion cues[2].

Since there is very little existing information available, the objectives of the investigation were quite broad. The primary aim was to establish whether or not a g-seat could provide useful heave motion onset cueing, and if so to identify the optimum drive algorithms. Secondary aims were to investigate the effects of possible vibration masking on heave cueing and to compare g-seat with and without platform motion cueing.

Physiology

Somatic Sensory System

Somatic cues are those which stimulate sensors in the body walls, as opposed to the head, limbs and internal organs. These sensors are either near the surface of the skin (cutaneous) or under the skin (sub-cutaneous), and comprise many different types of receptor, which are classified as either rapidly adapting (response dimishes in a fraction of a second) or slowly adapting (response continues for more than a second)[3]. They may also be described as displacement, velocity, acceleration or jerk sensitive depending on frequency response. The majority signal velocity or acceleration of the skin.

Kinaesthetic Sensory System

Kinaesthesia literally means a sense of movement, but is also usually taken to include static position sense. As used in this Paper it is defined as an awareness of positions and movements of limbs which is derived mainly from muscle receptors, but joint and skin receptors are also believed to contribute.

Humans are capable of sensing limb positions and movements, whether voluntary or imposed, with considerable accuracy. For example, detection thresholds of a fraction of a degree have been demonstrated[4]. The strength of the sensation depends on the rate at which the movement is imposed, so faster movements are more easily detected. Positional awareness is also good, and subjects have detected imposed movements of only a few degrees even when the rate of movement was very low (over a 15 minute period). However, we are much more aware of movement than we are of static position.

Heave Motion Effects

Heave motion cues are sensed by the pilot in a number of ways, including the vestibular organs. However, the effects which are most apparent to the pilot are those sensed via the somatic and kinaesthetic systems, as summarised below:

o As 'g' increases, the area of skin contact spreads as the body sinks into the seat. Conversely, contact with the shoulder and lap straps is reduced.

o The seat appears to become harder as more of the force becomes concentrated on the lower parts of the pelvic girdle.

o The angles of the hips, knees, shoulders and elbows change slightly as the body moves.

o The eye-point changes as the body sinks under positive 'g'.

o The body deforms inside and out, and the muscles tense as they resist.

o Blood pools in the legs.

Conventional motion systems stimulate some of these senses, but on a much reduced scale and for only brief periods. G-seats attempt to stimulate the somatic and kinaesthetic sensors directly, by simulating the first four effects above, without having to duplicate the actual accelerations.

Previous Research

By far the greatest amount of research on g-seats has been related to fixed-wing aircraft, almost all in the USA with pneumatic systems[5,6,7]. Since these seats are also used to provide cues in other axes using differential inflation and deflation of cushion cells, they are sometimes referred to as dynamic seats. The results of this research have been mixed but on balance, dynamic seats do appear to provide useful cueing. Early pneumatic systems were designed to provide sustained cues and had low bandwidth, but even so they were found to provide some onset cues[1,8]. Recent designs are much more responsive, allowing researchers to investigate onset cueing, though so far only rotational motion appears to have been investigated[9]. Perhaps the research closest in nature to the work described in this Paper is that of Ricard et al[10], who included a dynamic seat in a general assessment of simulator fidelity using a helicopter hovering task. An interesting finding was that the seat, which was configured to provide pitch, roll and surge cues (but not heave cues), affected height control. In addition, none of the pilots liked the seat cueing and believed that it degraded their performance, though in fact the opposite was true. A later experiment[11] found that there was no performance improvement in a ship approach and landing task, and again pilots did not like the seat cueing. Since the seats in these experiments provided cues in multiple axes, their contribution to vertical axis cueing is unclear.

In the UK, Cranfield Institute of Technology (CIT), under contract to RAE Bedford, adapted a conventional ejection seat to provide normal acceleration cues. This is a single-axis device, but it uses a hydraulic actuation system with inherently high bandwidth, as described in the next section. Trials with the seat were inconclusive, but pilot acceptance was good[12] and seats of essentially the same design are still being produced for flight training simulators. The work described here is the first application of this type of g-seat to helicopter simulation.

Description of G-seat

Hardware

The g-seat at RAE Bedford is a Martin-Baker Mk4 ejection seat which has been modified by CIT to provide 'g' cues. The lower half of the seat, or 'bucket', is driven vertically along guide-rails by a hydraulic actuator as shown in Fig 1. The top half of the seat, containing the back-rest, remains fixed with respect to the cockpit. Housed within the bucket is a 'pan', which consists of a U-shaped support as shown in Fig 2. This is driven vertically with respect to the bucket (and seat cushion) by a second hydraulic actuator, independent of the bucket actuator.

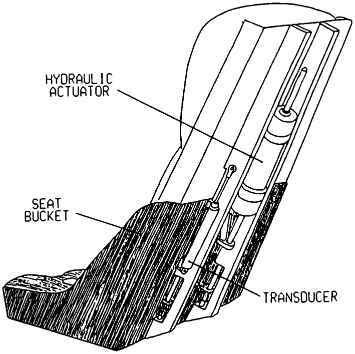


Fig. 1 Seat bucket

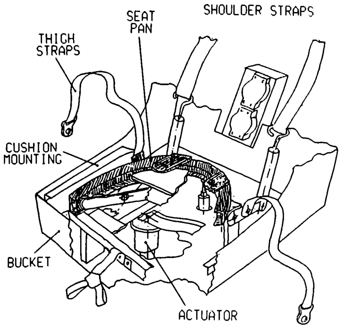


Fig. 2 Seat pan

Upward movement of the pan into the seat cushion provides the seat-hardening effect of positive 'g'. There is a limit to how far this can be taken however because the body begins to lose contact with the sides of the cushion (a negative cue). Bucket movement is the reverse of pan movement. Under positive 'g' it moves downwards, partly to negate the upward travel of the pan, but mainly to simulate changes in limb orientation and eye-height.

Linked mechanically to the pan, via tension-limiting springs, are two restraining straps which pass over the shoulders to the quick release connector (Fig 3). Also linked to the pan are two strans which pass over the thighs to the quick-release connector. A lost-motion device ensures that these straps are driven only over the lower half of pan travel, corresponding to negative 'g'.

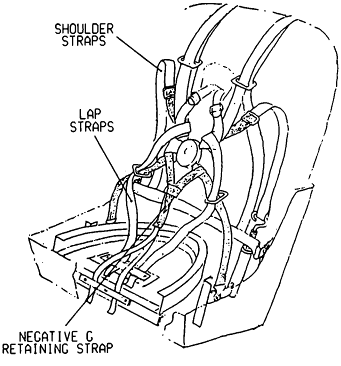


Fig. 3 Seat harness

Frequency responses of the seat show that for displacements of +/-5 mm, the pan has a bandwidth of 12 Hz (phase lag 90 deg) and the bucket a bandwidth of 3 Hz (phase lag 60 deg).

Algorithms

For simplicity, the pan and bucket were driven as linear functions of normal acceleration at the pilot's station. Two pan offsets, representing soft and hard seats, and two pan sensitivities (5 and 10 mm/g) were evaluated. In every case the bucket was driven to negate the changes in pilot eye-height caused by pan movement and, in addition, three bucket sensitivities (0,5 and 10 mm/g) were evaluated. Vibration was applied to the seat bucket by a computer controlled oscillator unit in the cockpit. The magnitude and frequency of vibration was determined by aircraft model parameters, namely airspeed, normal acceleration and rotor speed.

Description of Trial

Helicopter Mathematical Model

An idealised helicopter mathematical model, with rotational responses based on first and second order transfer functions, was used[13]. Briefly, its characteristics were as follows. At all speeds, pitch and roll cyclic commanded pitch and roll rates. Rudder pedals commanded sideslip above 40 kn, and yaw rate below 30 kn, blended linearly with airspeed between 30 and 40 kn. With pedals centred, sideslip was suppressed and yaw rate damped in the upper and lower speed ranges respectively. Throughout the flight envelope, cross-coupling between the rotational axes was eliminated. Although rotational handling and control were idealised, the performance of the model was representative of a Westland Lynx helicopter. Rotor calculations assumed a simple disc, but engine and downwash effects were fully modelled.

Simulator Facility

A generic cockpit was mounted on a Redifon motion platform which was free to move in heave (+/-0.3 m), pitch (+15, -10 deg) and roll (+/-10 deg). The cockpit was equipped with all the controls and displays necessary for a helicopter handling study, including a conventional centre-stick for cyclic control which was driven by a hydraulic digital control loading system. The collective lever was driven by a programmable electric system, and was configured as a conventional collective with adjustable friction. A Head Up Display (HUD) was used to allow pilots to fly 'eyes out' of the cockpit at all times and to compensate for simulator deficiencies, especially restricted field of view[13]. An attitude-based format was used as shown in Fig 4.

For most of the investigation, including all the formal assessments, the visual scene was presented to the pilot on a collimated 625-line monitor with a field of view of 48 deg in azimuth and 36 deg in elevation. The image was generated by a TV camera traversing a 700:1 scale model belt. A few sorties made use of a three-window Singer Link Miles Image 3T CGI system with 1000-line monitors. The field of view was approximately 120 deg in azimuth and

36 deg in elevation, plus an additional 12 deg of downward visibility in the side monitors.

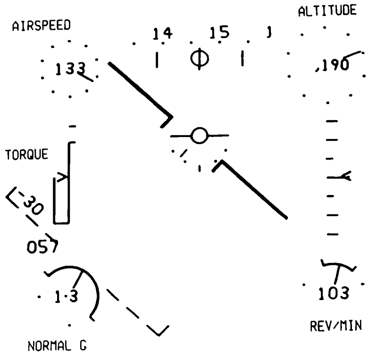


Fig. 4 Head Up Display Format

Pilot Background

A total of 14 pilots was used in the trial. Of these, 4 were qualified test pilots, 6 were test pilots nearing the end of their training, and 4 were squadron pilots. Their experience of simulators ranged from virtually none to considerable.

Task Description

The trial was intended to evaluate the g-seat as a 'manoeuvre' rather than as a 'disturbance' motion cueing device, and the tasks reflect this. Manoeuvre cues are an integral part of the pilot's control loop and indicate the effects of a deliberate control action. If the cues are different from those he expects, as in a simulator for instance, he may modify his control strategy. Disturbance cueing on the other hand alerts the pilot to uncommanded motion of his vehicle arising from outside the pilot's control loop, eg turbulence or failures, which he may then correct.

As part of a familiarisation exercise, pilots were free to fly, and comment on, any manoeuvres they wished. In addition, there were two formal tasks which were chosen to emphasise aggressive manoeuvring in the vertical plane, and these were assessed more objectively. In the first task, the pilot flew over a series of four in-line hurdles at 90 kn, staying below 15 m for as long as possible between each. In order to minimise coupling between pitch rotation and height, and to 'quicken' the height dynamic response, pilots were briefed to use collective to control height and pitch cyclic to control speed. The second task consisted of a 30 m bob-up from a stable hovering position, with the task being completed by lining up a chimney-top with the horizon. This task was as aggressive as aircraft limits allowed.

Results

General

From the beginning pilot comments were very encouraging, indicating that the seat was indeed providing a useful cue. Reassuringly, there was also general agreement about which of the drives felt most convincing. For aggressive flying, a scaling of 10 mm/g for both pan and bucket was chosen, but for more gentle manoeuvres a scaling of 5 mm/g (pan and bucket) was considered to be more appropriate. Based on these results, a nonlinear drive was eventually implemented, but for reasons of continuity the former was used in the formal assessments.

Unfortunately, only a few of the pilots had time to complete a full assessment, but all were able to compare seat-off and seat-on configurations. In summary, subjective comments revealed the following:

o Pilots were more comfortable with configurations where g-cueing was provided. Less concentration was required and confidence was increased, particularly at low level where they could fly more aggressively.

o The seat cannot provide all the effects of normal acceleration. There is no distension of the stomach, and in a steady-state situation any cues from the seat are easily dismissed. This was alleviated to some extent by the increased level of vibration at higher 'g', which is less easily dismissed.

o The simulator felt very deficient without g cues, but also lacked realism if vibration was missing. All the pilots preferred configurations where both g-cueing and vibration were provided, and most could not differentiate between platform motion on and off configurations.

o The rapid g-seat onset cue was liked, but this was a little too rapid for small manoeuvres.

o Cueing in turns, pull-ups and bunts was particularly good.

Hurdles Task

Recordings from this exercise confirmed pilot comments that over-controlling in the vertical axis was reduced when g-cues were provided (Fig 5). The flight path was less oscillatory between obstacles and pilots were able to fly lower. When seat cues were missing, they were very conscious of relying on the HUD more and were reluctant to move the collective rapidly.

An important aspect of this task was how well the pilot could recover to a level flight condition once he had cleared each obstacle. One pilot noted that he was more aware of pulling collective, while another commented that he could get lower because he felt he knew when to pull up. In an attempt to quantify this, the average times to ground impact at the point where pilots initiated the pull-up and the average rates at which collective was then applied, were derived. Time to impact was used as a measure because it

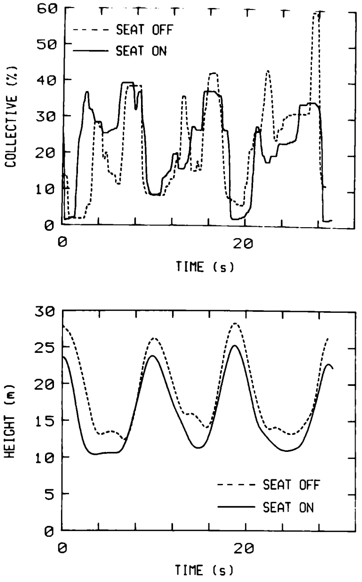


Fig. 5 Height control in hurdles task

combines the height and descent rate information used by the pilot to judge the pull-up. Other things being equal, one would expect a reduction in impact time to be accompanied by an increase in collective rate. However, with the g-seat in operation, not only are impact times reduced, but they are also accompanied by decreased collective rates. This is shown in Fig 6, on a graph of inverse collective rate vs impact time, as a general trend to the left and up from g-seat off to on. By using the inverse of collective rate, an approximately linear relationship with time to impact is revealed with g-seat on, showing a good degree of consistency between pilots. The seat-off data exhibit too much scatter to allow any relationship to be established.

The above comments, on being aware of pulling collective and knowing when to pull up, relate to the two components of the manoeuvre cue; the initial surge of acceleration and its subsequent washout due to aerodynamic heave damping. The initial cue was the most obvious to the pilots and was the major reason for the reduced collective movements observed. However, the fact that most pilots waited longer before pulling up indicates that they also perceived the decreasing descent rate due to heave damping. Heave damping contributes to the longer term response of the helicopter following a pilot control action, eg where the pilot is operating temporarily open-loop, but this is less clearly associated by the pilot with collective movement than the immediate response. In the phase of flight immediately prior to pull-up, the pilot is acting as a passive observer and his control activity is zero. Separated in time from other supporting cues, such as lever movement, the seat cues may be more prone to misinterpretation by pilots than in other phases of flight where they are operating more closed-loop. This could explain why two pilots (Nos 3 and 7 in Fig 6) pulled up sooner when the seat was operating.

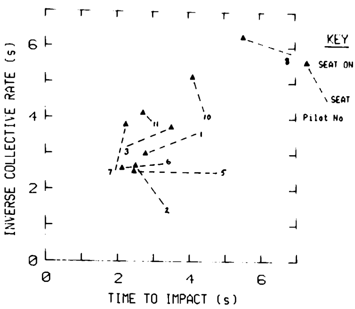


Fig. 6 Control behaviour in hurdles task

Bob-up Task

Pilot opinion was again very favourable. For example, one pilot reported that without seat cues this task was nowhere near as realistic, and another that he needed to refer more to instruments. Analysis of bob-up data also showed changes in pilot control consistent with improved perception of vertical accelerations.

The optimum bob-up manoeuvre in flight is characterized by pulse-like collective movements with approximately equal positive (C_1) and negative (C_2) magnitudes as shown in Fig 7a. The time intervals of these pulses are unequal due to aerodynamic heave damping effects, which result in the vertical velocity response approximating a first order lag with a time constant, τ, of 1.65 seconds, as shown by the broken line. The optimum timing ratio (T_2/T) also depends to some extent on the total time, T (which for this task was approximately 6.5 seconds), and is approximately 0.23. Clement et al[14] have shown that if heave damping is not perceived correctly by the pilot, due to a reduction in available cues, it is likely that the time intervals will tend to become more equal (ie $T_2/T = 0.5$), as if the aircraft response is perceived to be like a pure integrator (see Fig 7b).

In the present study, the differences in pilot background and the limited training time available produced some variation in piloting technique and levels of aggression. In order to allow for non-optimum use of collective, the timing ratio T_2/T was plotted against collective amplitude ratio (C_2/C_1) in Fig 8. Points near the upper boundary

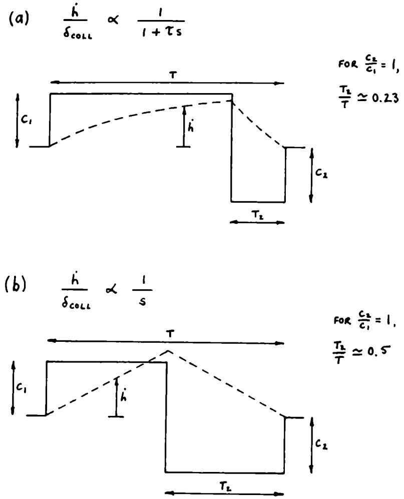

(a)
$\frac{\dot{h}}{\delta_{COLL}} \propto \frac{1}{1+\tau s}$
FOR $\frac{C_2}{C_1} = 1$,
$\frac{T_2}{T} \simeq 0.23$
T
C_1
$\dot{h}$
C_2
T_2
(b)
$\frac{\dot{h}}{\delta_{COLL}} \propto \frac{1}{s}$
FOR $\frac{C_2}{C_1} = 1$,
$\frac{T_2}{T} \simeq 0.5$
T
C_1
$\dot{h}$
C_2
T_2

Fig. 7 Effect of heave damping on collective timing in bob-up task

indicate poor perception of heave damping, whilst those near the lower boundary indicate good perception. Note that all but one pilot (No 3 in Fig 8) showed improvements in their perceptions of heave damping or in control technique (collective amplitude ratio) when g-seat cueing was provided. The exception was one of the two pilots who also behaved differently in the hurdles task. The other did not perform the bob-up task. Presumably the reasons for this are the same as discussed in the previous section.

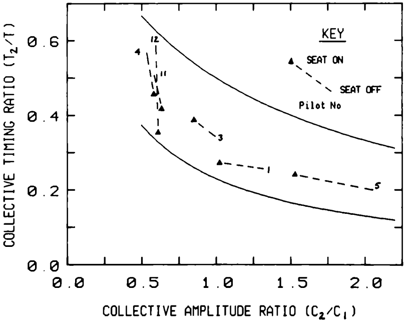


Fig. 8 Control behaviour in Bob-up task

Also of interest in this task was how quickly the pilot could achieve a precision hover following the gross manoeuvre. All the pilots felt that the collective activity was reduced when g-seat cueing was present, but the extent of the improvement varied considerably. For the development pilot (No 1), who was accustomed to performing this manoeuvre in flight and who was also familiar with the simulator, the difference was quite dramatic as shown in Fig 9.

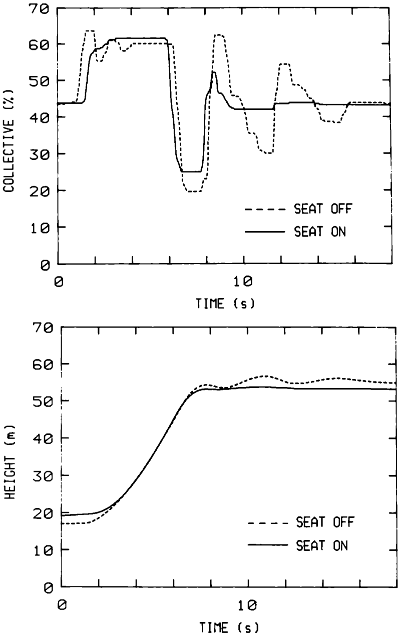


Fig. 9 Height control in bob-up task (simulator)

By way of comparison, a similar manoeuvre (of less magnitude) performed by the same pilot in flight is shown in Fig 10. Other pilots, though not showing such a great improvement, were able nevertheless to achieve better results when the seat was in use. Data indicate that the collective pull up at commencement of the precision hover phase was earlier and consequently less abrupt, and there was less overcontrol during the hover phase.

Hover

Although control activity was reduced when using the g-seat, as shown by the Power Spectral Density (PSD) of collective displacement in Fig 11, there was still a significant oscillation in the pilots' collective movements which does not occur in flight.

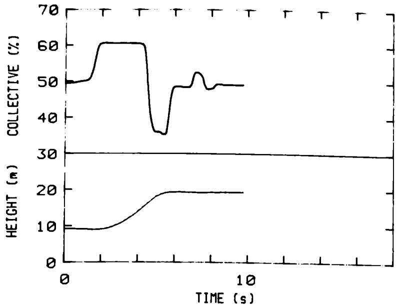


Fig. 10 Height control in bob-up task (flight)

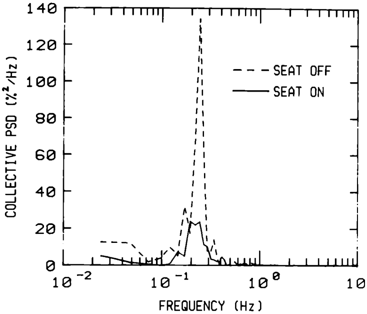


Fig. 11 Control activity during hover (simulator)

In an attempt to determine the cause, a number of possible contributory factors were considered. The pilot is attempting to control height with what is essentially an acceleration controller, ie two integrations removed. This is known to require significant compensation by the pilot[15], which he is able to provide in flight by making full use of the motion and visual cues available to him. In the simulator, heave-motion cues are very weak and visual cues are lacking in three respects:

- The field of view is limited.
- The TV height servo is being operated at very low signal levels where friction becomes significant.
- The TV monitor resolution (625 lines) does not allow low vertical velocities to be detected early enough by the pilot.

Taken together, and bearing in mind the pilot comments about 'g' cueing being too aggressive, a seat drive based on vertical velocity was investigated. This would have the advantage of cueing the pilot to vertical velocity changes before these can be detected visually, in effect providing a substitute cue. When implemented, this made hovering, if anything, slightly easier than in the aircraft, with comparable collective activity as shown in Fig 12. Although these findings are predominantly of academic interest, very little seat movement (less than 1 mm peak to peak) was required and was subtle enough to prevent it being perceived as a false cue. The effect was also observed when using CGI visuals.

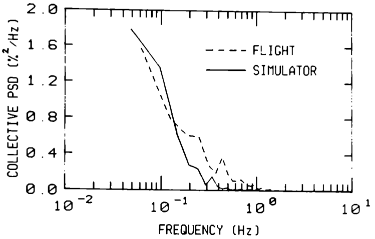


Fig. 12 Comparison of control activity in hover

Discussion

Learning effects are frequently a problem for researchers due to the limited time available and the differing backgrounds and skills of the pilots. For the purposes of this paper however, the learning process was itself of interest due to the potential training application of the work. Indeed, first impressions were regarded as extremely important since pilots quickly adapt (often subconsciously) to the simulator's limited cueing environment. Although task performance was obviously affected by learning, this was alleviated by pairing configurations of most interest together for direct comparison and by changing the order of presentation to various pilots.

The level of aggression used by pilots is an important factor in determining whether or not they are likely to find motion cues beneficial. A pilot who is prepared to tolerate greater errors or reduced performance can operate at lower 'gain', in the context of the overall pilot-aircraft control loop, where the cues obtained visually may well be sufficient. Not all the pilots flew as aggressively as intended, but nevertheless they all liked the g-seat cues and felt that control activity and performance were more like flight.

A few pilots stated during familiarisation that the cues provided by the seat seemed ambiguous or even reversed. Perhaps the sensation of losing contact with the edges of the cushion was sufficient to confuse or overcome the sensation of seat-hardening for these pilots. In the formal tasks this appeared to be less of a problem, although two pilots did show some signs of negative cueing. This might indicate either that they had quickly adapted to the seat cues or that the seat cues seemed more natural in conjunction with all the other cues available. The latter suggests that the seat provides good manoeuvre cues, which confirm the magnitude and timing of a pilot-commanded manoeuvre, but may not be so useful for disturbance cueing, where the direction of motion must be deduced solely from seat cues if the pilot is to benefit from them. Disturbance motion cueing is likely to be the subject of future research.

Conclusions

The literature[3,4] indicate that g-seats ought to be more effective for transient cueing than for sustained cueing, since we are generally much more aware of rapid movements than we are of static position, and this was found to be the case. Although the seat does give sustained 'g' cues, pilots' comments indicate that these are quite weak. However, the variation of vibration amplitude with normal 'g' enhances the sustained cues since vibration is less easily ignored than seat pressure or body orientation.

G-seats should be regarded as complementing motion platforms rather than substituting for them. They have been shown by this study to be effective at providing cues in the axis where platform cues are weakest. In conjunction with a synergistic motion platform, the use of a g-seat would allow a reduction in heave motion displacement, enabling motion cueing in other axes to be enhanced.

There is little doubt that g-seat cueing enhances the realism, and hence pilot acceptance, of piloted helicopter simulation. Unlike conventional platform motion cues, which are sensed subconsciously by the vestibular system, g-seat cues are readily apparent to the pilot, so that he is fully aware of whether or not they are helping him. All the pilots who took part in this study believed that the cues were beneficial, and the recorded data on control activity and performance support that belief. The experiment with vertical velocity cueing in hover, although largely of academic interest, was a convincing demonstration of the strength of the somatic and kinaesthetic senses.

References

1	G Kron L Young W Albery	High-g Simulation - The Tactical Aircraft Simulator Problem. NAVTRAEQUIPCEN IH-294 (1977)
2	Heffley Clement et al	Determination of Motion and Visual System Requirements for Flight Training Simulators. US Army Report No 546 (1981)
3	K R Boff L Kaufman et al	Handbook of Perception and Human Performance Vol 1, Chapter 12.
4	K R Boff L Kaufman et al	Handbook of Perception and Human Performance Vol 1, Chapter 13
5	G J Kron	Motion Simulation Enhancement: The Development of a Research g-seat System. NAVTRAEQUIPCEN IH-226 (1973)
6	B R Ashworth B T McKissick D J Martin	Objective and Subjective Evaluation of the Effects of a g-seat on Pilot/Simulator Performance during a Tracking Task. NAVTRAEQUIPCEN IH-294 (1977)
7	R V Parrish B T McKissick B R Ashworth	Comparison of Simulator Fidelity Model Predictions with in-simulator Evaluation Data. NASA-TP-2106 (1983)
8	W B Albery D R Gum G R Kron	Motion and Force Cueing Requirements and Techniques for Advanced Tactical Aircraft Simulation. AGARD-CP-249 (1978)
9	E A Martin R K Osgood G R McMillan	The Dynamic Seat as an Onset Cueing Device. AIAA 87-2438-CP (1987)
10	G L Ricard R V Parrish et al	The Effects of Various Fidelity Factors on Simulated Helicopter Hover. NAVTRAEQUIPCEN IH-321 (1981)
11	R T Galloway	Helicopter Training Research at the Visual Technology Research Simulator. Royal Aeronautical Society Proceedings (1984)
12	N O Matthews C A Martin	The Development and Evaluation of a 'g' Seat for a High Performance Military Aircraft Training Simulator. AGARD-CP-249 (1978)
13	S L Buckingham G D Padfield	Piloted Simulations to Explore Helicopter Advanced Control Systems. RAE Technical Report 86022 (1986)
14	W F Clement W B Cleveland D L Key	Assessment of Simulation Fidelity using Measurements of Piloting Technique in Flight. AGARD-CP-359 (1984)
15	D T McRuer E S Krendel	Mathematical Models of Human Pilot Behaviour. AGARD-AG-188 (1974)

89-3275-CP

MATRIX AND VECTOR DATA STRUCTURES IN THE SIMULATION LANGUAGE ADSIM

Aslaug Haraldsdottir
Applied Dynamics International
Ann Arbor, Michigan

Abstract

Matrix and vector data structures have been introduced into the simulation language ADSIM, which was originally designed and optimized for the AD 100 real–time simulation computer using only scalar variables. The powerful operator notation, allowing both conventional arithmetic and element–by–element operations is discussed. Matrix and vector arithmetic provides very compact and readable notation, but the issues of computational efficiency are also treated in this paper. Finally, the introduction of matrix and vector data structures has made it possible to extend the ADSIM language into the areas bordering simulation, such as dynamic analysis and control system design. The additional functionality in the standard ADSIM library is discussed in this paper, along with some examples to show its use.

1 Introduction

The introduction of vector and matrix data structures into a simulation language is discussed in this paper. The particular language described is ADSIM for the AD 100 real–time simulation computer. The AD 100 computer and ADSIM were designed and optimized to perform scalar computations and numerical integration with execution speed comparable to supercomputers. Version 7.0 of ADSIM includes vector and matrix data structures and allows extensions of the language into areas outside dynamic simulation.

The objective of this paper is to introduce the design and functionality of ADSIM V7.0. The paper emphasizes how the language was extended with the goal of maximizing execution speed for typical simulation applications. This was done while allowing very readable notation using operators such as * for matrix and vector products. The paper also discusses the additional functionality in areas such as dynamic alalysis and control system design that is now possible in ADSIM. Several numerical analysis algorithms implemented in the standard ADSIM library are discussed.

The paper is organized such that section 2 introduces the basic operator notation allowed in ADSIM for matrix and vector data structures. Section 2 also discusses computational efficiency for typical simulation applications and the representation of dynamic equations in ADSIM. Section 3 introduces the additional functionality such as system linearization, eigenvalue solution, singular value decomposition and nonlinear equation solution now available in ADSIM. Examples are then presented in section 4, showing the power of the operator notation and some results on execution times.

2 Matrix and Vector Use in ADSIM

In real–time simulation of large systems it is of utmost importance that both software and hardware be used to achieve maximum execution speed. This was the major objective in the design and implementation of matrix and vector arithmetic for the AD 100. One of the reasons for slower execution times for matrix and vector variables are address calculations to access memory locations and the overhead associated with do–loop constructs. A second objective for the language design was to allow as much "mathematical" notation as possible within the constraints of using the ASCII character set. This section describes the features in ADSIM that are especially designed to avoid computational inefficiencies and provide readable notation.

In light of the above stated objectives, ADSIM allows direct operator notation for addition, subtraction and multiplication of matrices and vectors of compatible dimensions. Thus the arithmetic expression

$$A = B + C * D \tag{1}$$

is accepted for any compatible scalars, vectors and matrices. To maximize computing speed, the arithmetic implied by $+, -, *$ is implemented in AD 100 machine language, eliminating the need for loops to perform these common operations. Table 1 shows the definition of conventional arithmetic operators in ADSIM.

$\hat{}T$	$A = B\hat{}T$	$A, B : n \times m$	Transpose
$*$	$A = s * B$	s: scalar	$a_{ij} = sb_{ij}$
		$A, B : n \times m$	
	$A = B * C$	$B : n \times r$	Matrix
		$C : r \times m$	multiplication
		$A : n \times m$	
$/$	$A = B/s$	s: scalar	$a_{ij} = b_{ij}/s$
		$A, B : n \times m$	
$+$	$A = B + C$	$A, B, C : n \times m$	$a_{ij} = b_{ij} + c_{ij}$
$-$	$A = B - C$	$A, B, C : n \times m$	$a_{ij} = b_{ij} - c_{ij}$

Table 1: Matrix/Vector Conventional Arithmetic

In most continuous system simulations the bulk of the computational task involves scalar calculations. However, there are systems that are much more conveniently modeled using matrix and vector notation. The arithmetic described in Table 1 is very useful in these problems, but may invite considerable inefficiency since most physical system models lead to sparse matrices. Discretized models of continuous systems such as beams and fluid flow are good examples of sparse systems where matrix notation is essential.

To allow the clarity of operator notation for sparse problems, yet avoid unnecessary operations such as multiplication and addition of zeros, ADSIM has a wide range of elemental operations. An example is the expression

$$A = B + C\{*\}D + \text{esin}(E) \quad (2)$$

which is equivalent to the mathematical notation

$$a_{ij} = b_{ij} + c_{ij}d_{ij} + \sin(e_{ij}) \;\; i = 1, ..., n \; j = 1, ..., m \quad (3)$$

Table 2 shows the notation for elemental operations, where { } around the operator implies that the operation is to be performed on the elements of the vectors and matrices as if they were a collection of scalars.

$**$	$A = B\{**\}s$	s: scalar	$a_{ij} = b_{ij}^{s}$
		$A, B : n \times m$	
	$A = s\{**\}B$	$B : n \times m$	$a_{ij} = s^{b_{ij}}$
		s: scalar	
	$A = B\{**\}C$	$A, B, C : n \times m$	$a_{ij} = b_{ij}^{c_{ij}}$
$*$	$A = B\{*\}C$	$A, B, C : n \times m$	$a_{ij} = b_{ij}c_{ij}$
$/$	$A = B\{/\}C$	$A, B, C : n \times m$	$a_{ij} = b_{ij}/c_{ij}$
$+$	$A = s\{+\}B$	s: scalar	$a_{ij} = s + b_{ij}$
		$A, B : n \times m$	
$-$	$A = s\{-\}B$	s: scalar	$a_{ij} = s - b_{ij}$
		$A, B : n \times m$	

Table 2: Matrix/Vector Element Arithmetic

In addition to the operators listed in Table 2, all the ADSIM intrinsics such as trigonometric and exponential functions are allowed to operate element–by–element on matrix and vector arguments when prefixed by "e" as in equation (2). All the operations listed in Table 2 and all the intrinsic functions in ADSIM are implemented using machine language for maximum computational speed.

Example 1 in section 4 shows the use of operator notation for vectors and matrices in ADSIM and some results on execution times.

ADSIM is a simulation language designed for real–time simulation and includes features for the use of state variables and their numerical integration. Twelve fixed–step integration algorithms are provided and the language has been extended to allow state vectors to be automatically integrated as a collection of scalar variables. Vector and matrix data structures are not used in the implementation of the standard integration algorithms.

A dynamic model to be simulated using ADSIM is expressed as a collection of first–order differential equations

$$x' = f(x, u, t) \quad (4)$$

where x is the state–variable (scalar or vector) and u denotes input functions. A typical system will include many nonlinearities in the functions f and thus will not be represented in its entirety using vector or matrix data structures. However, parts of the dynamic equations may be linear in the variables x and u such that they are conveniently written on the form

$$x' = Ax + Bu \quad (5)$$

where x and u are vectors and A and B are appropriatly dimensioned matrices. Another commonly occurring form of differential equations arises from multibody dynamics

$$M(q)q'' + C(q, q') + G(q) = F(t) \quad (6)$$

where q is the vector of generalized coordinates, M is the inertia matrix and F is a vector of external forces. C and G are nonlinear functions of q and q'. To convert the system in (6) to first–order form, the mass matrix M needs to be inverted every time the state variable derivatives are calculated. The first–order form of (6) is

$$\begin{aligned} v' &= w \\ w' &= M(v)^{-1}\left[F(t) - C(v, w) - G(v)\right] \end{aligned} \quad (7)$$

where $q = v$ and $q' = w$.

The matrix inversion indicated in (7) should be implemented as the solution of a set of linear equations

$$Mw' = b \quad (8)$$

ADSIM supplies functions to solve (8) using LU factorisation. They are implemented using routines based on the Basic Linear Algebra Subroutines (BLAS) described in [2]. The BLA routines are implemented in AD 100 machine language to achieve maximum execution speed.

3 Matrix Functionality

Earlier versions of the simulation language ADSIM allowed only scalar variables and some special data structures to perform function interpolation and collection of simulation results. This implied that the language was not well suited to perform dynamic analysis or control system design calculations. With the addition of vector and matrix data structures, there is the potential to extend the ADSIM simulation tool into entirely new domains of numerical analysis. The first release of ADSIM with vectors and matrices includes a collection of numerical analysis functions and they will be described here.

The assumptions made in the design and selection of numerical analysis tools for ADSIM are that with the exception of LU factorization, the functions are unlikely to be used as part of the time–critical integration loop. Therefore, minimum execution speed is not considered a primary objective. Selection of algorithms is made based on issues such as ease of implementation in ADSIM, robustness on a variety of problems and ease of use.

To start extending the ADSIM language into new areas of numerical analysis, solutions of problems related to simulation were first considered. It is often of interest to investigate the eigenvalues of a model, one of the reasons being difficulty in obtaining a stable fixed–step integration. In order to do that for a nontrivial model, the equations of motion have to be linearized around an operating point to obtain an approximation on the form of (5). The eigenvalues of the matrix A then give the time constants or frequency and damping ratio of the modes of the system in the neighborhood of the operating point. The eigenvalues with the largest magnitude are the ones that may cause stability problems in the numerical integration [3].

Another problem of interest in simulation is finding a steady–state or equilibrium solution for a model. In aircraft simulation this is often referrend to as "trimming." This in general involves solving the problem

$$f(x) = 0 \tag{9}$$

where x and f are a subset of the variables and equations in (4). The functions f are usually nonlinear and (9) may have multiple solutions.

A number of linear algebra problems that are increasingly being used by control system designers depend on the singular value decomposition for their solution. Matrix norms, rank and nullspace of matrices can be analyzed using the singular value decomposition. Questions of stability robustness of control systems are also addressed using the SVD as the basic tool.

The numerical analysis areas mentioned above will be covered in the early releases of ADSIM with matrix and vector notation. Algorithm selection and implementation of each is described in sections 3.1–3.4.

3.1 Linearization of a Model

Assume that the dynamic model to be linearized is represented on the form

$$x' = f(x, u) \tag{10}$$

where $x \in R^n$ is the state vector, $u \in R^m$ is the input vector and $f : R^n \times R^m \rightarrow R^n$. It is desired to find a linear approximation to the relationship (10) on the form (5) that is valid in a small region around a point $x = x_0$, $u = u_0$. Referring to (5)

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial f_n}{\partial x_1} & & \frac{\partial f_n}{\partial x_n} \end{bmatrix}_{x=x_0} \tag{11}$$

$$B = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \cdots & \frac{\partial f_1}{\partial u_m} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial f_n}{\partial u_1} & & \frac{\partial f_n}{\partial u_m} \end{bmatrix}_{u=u_0} \tag{12}$$

The partial derivatives indicated in (11) and (12) are usually not available analytically and thus numerical evaluation is necessary. The derivatives are replaced with finite differences such that

$$\frac{\partial f_i}{\partial x_j} \simeq \frac{\Delta f_i}{\Delta x_j} \tag{13}$$

The implementation of the calculations proposed in (11–13) is simple, but the selection of the size of Δx_j is not obvious and may have a major influence on the accuracy of the derivatives. The two factors in choosing Δx_j are to keep it small enough that the functions f are approximately linear in the range $x_0 \pm \Delta x$ but not so small that roundoff errors dominate the subtractions performed in (13). Using these two considerations it is possible to write a perturbation selection scheme to evaluate (11–13) accurately, and this was done in [4].

The algorithm in [4] is quite elaborate and therefore a straightforward implementation was made in ADSIM using information about the magnitude of the elements of x_0 to scale the perturbations.

Example 2 shows the use of the linearization function in ADSIM.

3.2 Eigenvalue Solution

ADSIM incorporates two methods to solve the eigenvalue problem; one solves a symmetric form and the other a general real matrix. These functions can be applied to any square matrix, but would often be used to analyze the matrix A obtained from (11) by numerical linearization.

- The symmetric eigenvalue problem solved in ADSIM is stated as

  $$(M\lambda_i - K)x_i = 0 \tag{14}$$

  This form of the eigenvalue problem may be derived from differential equations of the form

  $$M\ddot{x} + Kx = 0$$

  whete M and K are symmetric positive definite matrices. The problem (14) is solved in ADSIM using the power method and Hotelling deflations [5]. The eigenvalues and vectors are all real and convergence is rapid if all eigenvalues are widely spaced. The method will fail to converge if some eigenvalues are repeated and converges slowly if some eigenvalues are closely spaced.

- ADSIM also solves the eigenvalue problem stated as

  $$(A - \lambda_i I)x_i = 0 \tag{15}$$

  where A is a general real matrix of dimension $n \times n$. The algorithm used to solve (15) applies double–shift QR iterations [6]. Eigenvalues may be complex and repeated. Before application of the QR–Algorithm, the matrix A is transformed to upper Hessenberg form using Householder reflections. Two ADSIM functions are based on this algorithm, one returns only the eigenvalues of A, the other returns its eigenvectors also.

3.3 Singular Value Decomposition

The singular value decomposition involves rewriting a matrix A as

$$A = U\Sigma V^T, \quad \Sigma = \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \tag{16}$$

where A is $m \times n$, U is $m \times m$, V is $n \times n$, Σ is $m \times n$ and D is a diagonal $r \times r$ matrix whose strictly positive elements are the singular values of A. The algorithm used in the ADSIM function involves first bidiagonalizing the matrix A by Householder reflections and then performing double–shift QR–iterations to obtain the singular values. A good description is given in [7].

ADSIM functions to find rank, condition number and the pseudo–inverse of a matrix are also available, all assume that a singular value decomposition of the matrix has been performed.

3.4 Nonlinear Equation Solution

Finding the equilibrium of a set of differential equations on first order form is one of the problems that can be formulated as finding a vector x that satisfies

$$f(x) = 0 \tag{17}$$

where f is a set of n nonlinear functions in the n variables x. The problem (17) is a difficult one, the main source of difficulty being the possibility of multiple solutions. ADSIM provides two very different methods to solve (17), Broyden's method and a homotopy method.

3.4.1 Broyden's method

Broyden's method [8] is based on Newton's method to solve (17), but uses an approximation to the Jacobian of f. The algorithm consists of calculating an update to the current value x_k as

$$x_{k+1} = x_k + s_k$$

where

$$s_k = -A^{-1}f(x_k)$$

The next value of A is then calculated as

$$A_{k+1} = A_k + \frac{(y_k - A_k s_k)s_k^T}{s_k^T s_k}$$

where

$$y_k = f(x_{k+1}) - f(x_k)$$

The matrix A in Broyden's method takes on the function of the Jacobian f' in Newton's method. To start the algorithm, an initial value A_0 needs to be obtained. An initial finite difference approximation to the Jacobian is used as A_0.

3.4.2 Homotopy method

Homotopy methods for solving the problem (17) involve defining a special function, called a homotopy function, $H(x,t): R^{n+1} \rightarrow R^n$ which has the original

n variables x and an extra one t. H is constructed such that for $t = 0$ and $t = 1$

$$\begin{aligned} H(x,0) &= E(x) \\ H(x,1) &= f(x) \end{aligned}$$

where $E(x)$ is a simple function for which we know that $E(x^0) = 0$. Therefore, at $t = 0$

$$H(x,0) = 0$$

has a solution x^0 that we know, and at $t = 1$

$$H(x,1) = 0$$

has a solution x^* which we are looking for. In general for arbitrary t, $x(t)$ solves

$$H(x(t),t) = 0 \tag{18}$$

and our objective is to follow the path $x(t)$ as t varies from 0 to 1.

Two common forms of the function $H(x,t)$ are the so–called Newton homotopy and fixed–point homotopy.

Newton homotopy uses a function H of the form

$$H(x,t) = f(x) - (1-t)f(x^0) \tag{19}$$

The start of the path, $t = 0$, is obvious here since $H(x,0) = f(x) - f(x^0)$ and any point x^0 will make $H(x,0) = 0$.

Fixed point homotopy uses a function H of the form

$$H(x,t) = (1-t)(x - x^0) + tf(x) \tag{20}$$

Here again, we can select any point x^0 to start the path.

In general, for nonlinear functions, the choice of starting point, x^0, will influence which solution of $f(x)$ the path leads to. The choice of $H(x,t)$ will also influence the progress toward that solution.

The problem of solving (17) is therefore to follow a path in the $n+1$ dimensional, $x(t), t$ space (i.e. the extended solution space). The path, as noted above, starts at $(x^0, 0)$ and (hopefully) terminates at $(x^*, 1)$. All points on this path satisfy (18). There are a number of ways to follow a path in the $(x(t), t)$ space [9]. The most common approach belongs to the class of predictor–corrector continuation methods. Let $z = (x(t)^T, t)^T$ and consider the initial value problem

$$H_z(p)\dot{z}(p) = 0, \quad z(0) = (x^0, 0) \tag{21}$$

where the differentiation is with respect to the path parameter p that denotes arc length. Therefore, x and t may both be viewed as implicit functions of the path parameter. It is easy to verify that equation (18) represents an integral curve of (21). We shall assume that the homotopy function is smooth and the Jacobian of $H(x(t),t)$, also the coefficient matrix in (21), has rank n. Under these assumptions, equation (21) generates nice paths in the extended solution space. Integration of (21) numerically is precisely what we mean by "following the path". Before integrating (21), we will need the derivative vector values, $\dot{z} = dz/dp$. System (21) is underdetermined and thus has an infinite number of solutions for $\dot{z}$. To uniquely determine the derivative vector, (21) is augmented with the following constraints

$$\dot{z}^T\dot{z} = 1, \quad \dot{z}^T\dot{z}_{last} > 0 \tag{22}$$

Equations (21) and (22) now yield a unique solution to $\dot{z}$. The second of the two constraints in (22) assures that a constant direction of traversing the path is maintained. The solution of (21–22) for $\dot{z}$ can be obtained using Gaussian elimination with pivoting. When integrating (21), especially with a large step size, errors are introduced such that the result does not satisfy (18) exactly. The integration procedure is therefore corrected by using Newton iteration steps to solve $H(z) = 0$. This is done using

$$\begin{bmatrix} H_z(z^k) \\ \dot{z}^T \end{bmatrix}(z^{k+1} - z^k) = \begin{bmatrix} -H(z^k) \\ 0 \end{bmatrix}; \quad k = 0,1,2... \tag{23}$$

Iterating (23) until convergence is attained yields a point on the path.

The problem of following the path satisfying (18) thus involves integration as a "predictor" step and the Newton iterations as a "corrector" step. The Newton iterations will converge if the predictor step yields a point sufficiently close to the path. If the Newton iterations do not converge after a certain number of iterations, the last integration step is repeated with a smaller step size. After a successful corrector step, a predictor step is performed and so on. This cycle is repeated until the last component of vector z is approximately 1.

To perform the predictor step, any integration scheme may be used. The Euler method is attractive because of its simplicity and has been recommended widely.

The success of the homotopy method when applied to nonlinear equations depends on the ability of the path to penetrate the $t = 1$ plane. This is virtually impossible to ensure for practical problems. Another requirement for success is that the matrix H_z be full rank everywhere in the solution space of (18). This condition is shown in [9] to be almost always satisfied.

4 Examples

Example 1 A simple example to show how powerful the operator notation shown in Tables 1 and 2 is on a linear system of equations is shown here. Consider the discretized model of a flexible structure described as

$$\begin{aligned} n'' + Dn' + Kn &= Bu \\ y &= Cn \end{aligned} \qquad (24)$$

where n is a vector of modal displacements, u is a vector of external forcing functions, and y is a vector of measured displacements. The matrices D, K, B and C are of appropriate dimensions. D and K represent viscous damping and stiffness coefficients and are of the form

$$\begin{aligned} D &= \mathrm{diag}\{2\xi\omega_1, ..., 2\xi\omega_m\} \\ K &= \mathrm{diag}\{\omega_1^2, ..., \omega_m^2\} \end{aligned} \qquad (25)$$

where m is the number of modes. The ADSIM program shown below represents the system described in (24–25). Note that the program takes advantage of the elemental operators in ADSIM to avoid full matrix multiplication with the diagonal matrices D and K.

```
TITLE "Flexible Structure"

MATRIX B(44,2),C(21,44)
VECTOR Y(21),W(44),U(2),
       D(44),K(44),
       OM(2),AMP(2)
VECTOR N(44),ND(44)
DATA   B=(@'bdata.dat'),
       C=(@'cdata.dat'),
       W=(@'wdata.dat')
DATA   OM(1)=68.8318, OM(2)=31.4159,
       AMP(1)=(10.0,20.0),
       zeta=0.001
REGION initial
! Set up modal damping and stiffness matrices
! in vectors, since they are diagonal.
! They are used in elemental operations.
    D  =  2*zeta*W
    K  =  W{*}W
END REGION
DYNAMIC continuous
! Control point forcing functions
! Part of the vector expression uses
elemental
! operations, denoted by {}
  U = AMP {*} ESIN(OM*system_time)
! Sensor point displacements:
  Y = C*N
! Modal accelerations
  ND' = B*U - D{*}ND - K{*}N
! Modal velocities
  N' = ND
END DYNAMIC
RUNSPECS endtime=1.0, speedup=1.0
EXECUTE 'continuous'
```

This example was previously programmed in ADSIM using scalar variables and about 900 lines of code. However, even though the code takes full advantage of operator notation and thus AD 100 micro code, the matrix version executes about 29% slower than the scalar implementation.

Example 2 As an example of the use of some of the new ADSIM functions, consider the dynamic model of an inverted pendulum on a cart as

$$(m_c + m)\ddot{x} + ml\cos\theta\ddot{\theta} - ml\dot{\theta}^2\sin\theta = f \qquad (26)$$

$$(J + ml^2)\ddot{\theta} + ml\cos\theta\ddot{x} - mgl\sin\theta = 0 \qquad (27)$$

Here, m_c is the mass of the cart, m, J and $2l$ are the mass, moment of inertia and length of the pendulum, x is the cart displacement and θ is the pendulum angular displacement clockwise from the vertical. f is a horizontal force applied to the cart. The model in (26–27) can be written using matrix and vector notation as

$$\begin{bmatrix} m_c + m & ml\cos\theta \\ ml\cos\theta & J + ml^2 \end{bmatrix} \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} - \begin{bmatrix} ml\dot{\theta}^2\sin\theta \\ 0 \end{bmatrix} - \begin{bmatrix} 0 \\ mgl\sin\theta \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \qquad (28)$$

An ADSIM program implementing the equations of motion in (28) is shown below. The pendulum balancing problem is open–loop unstable and we will not go any further in designing a stabilizing controller here. The ADSIM code shows how one might start addressing that problem, by expressing the equations of motion and obtaining a linearized state matrix. After the ADSIM program listing, the results of the linearization function invocation giving the state matrix A are shown. The initial conditions on the state vector

```
X
```

in the ADSIM code default to zero and this is the point about which we linearize.

```
TITLE "Pendulum on a Cart"
MATRIX M(2,2),A(4,4)
VECTOR C(2),G(2),F(2),
       YP(2),VP(2),Y(2),V(2),X(4)
REGION initial
    J            = mp*l*l/3
    M(1,1)       = mc+mp
    M(2,2)       = J +mp*l*l
    C(2)         = 0
    G(1)         = 0
! Obtain a linearized state matrix
    A,X@ = LINEARIZE(X,X',continuous,frac)
END REGION
DYNAMIC continuous
```

```
     sin_v2,cos_v2 = SIN_COS(V(2))
     M(1,2)        = mp*l*cos_v2
     M(2,1)        = M(1,2)
     C(1)          = -mp*l*Y(2)*Y(2)*sin_v2
     G(2)          = -mp*gravity*l*sin_v2
! The equations of motion are on vector
! form
     YP  = SOLVE(M,-C-G+F)
     VP  = Y
     Y,V = PARTITION_ROW(X)
! Combine Y and V in one state vector X
     X'  = AUGMENT_ROW(YP,VP)
END DYNAMIC
DATA mp     = 1,
     mc     = 10,
     gravity= 10,
     l      = 0.5,
     F      = (0.0,0.0)
DATA frac   = 1e-7
EXECUTE 'continuous'
```

```
Opened Log File 22-JUN-89 14:23:23 Interact
V7.0
AD100> s 2 stick
AD 100 console 2 attached
The AD 100 system file is loaded
The IOCP system file is loaded
Created: 22-JUN-89 13:41:30 ADSIM      V7.0
Pendulum on a Cart
EXECUTE continuous  V7.0b  17 June 1989
No graphics options installed
Single runmode, realtime environment, padded
frame timing.
frametime       84.6000E-6
steptime        84.6000E-6   /S
speedup          1.0000
endtime          1.0000
Type GO HELP for some helpful information.
AD100> !
AD100> ! Perform calculations only up to the
AD100> ! first integration
AD100> !
AD100> go check
Check runmode, realtime environment, padded
frame timing.
The AD 100 is suspended.
Suspended solution for static/step check.
AD100>> !
AD100>> ! Display the state matrix A
AD100>> !
AD100>> data a
A (1,1) =      0.0000
A (1,2) =      0.0000
A (1,3) =      0.0000
A (1,4) =  -731.7075E-3
A (2,1) =      0.0000
A (2,2) =      0.0000
A (2,3) =      0.0000
A (2,4) =     16.0976
A (3,1) =      1.0000
A (3,2) =      0.0000
A (3,3) =      0.0000
A (3,4) =      0.0000
A (4,1) =      0.0000
A (4,2) =      1.0000
A (4,3) =      0.0000
A (4,4) =      0.0000
AD100>> exit
Logging stopped 22-JUN-89 14:25:13 Interact
V7.0
```

5 Conclusions

This paper has shown how the simulation language ADSIM, originally designed and optimized for scalar calculations, has been augmented with matrix and vector data structures and very powerful notation for common arithmetic operations. The issues of computational efficiency have been discussed and it was shown that along with mathematical notation, the implementation of matrix and vector arithmetic using machine language was used to maximize execution speed.

It was also shown how the simulation language ADSIM can now be extended into new areas such as dynamic analysis and control system design, thus making the AD 100 real–time simulation computer a considerably more powerful tool for dynamic system analysis and design.

References

[1] ADSIM Reference Manual V7.0, Applied Dynamics International, Ann Arbor, Michigan.

[2] Dongarra, J.J., Bunch, J.R., Moler, C.B. and Stewart, G.W., *LINPACK Users' Guide*, SIAM, 1979.

[3] Haraldsdottir, A. and Howe, R.M., 'Multiple Frame Rate Integration,' Proceedings of the AIAA Flight Simulation Technologies Conference, Atlanta, Georgia, Sept. 7–9, 1988.

[4] Salane, D.E., 'Adaptive Routines for Forming Jacobians Numerically,' Sandia Report SAND86–1319, Sandia National Laboratories, Albuquerque, New Mexico, August 1986.

[5] Meirovitch, L., *Computational Methods in Structural Dynamics*, Sijthoff & Noordhoff, 1980.

[6] Wilkinson, J.H. and Reinsch, C., *Handbook for Automatic Computation,* Vol. 2: *Linear Algebra*, Springer Verlag, Berlin, 1971.

[7] Garbow, B.S., Boyle, J.M., Dongarra, J.J., Moler, C.B., *Matrix Eigensystem Routines-EISPACK Guide Extension*, Springer–Verlag, Berlin, 1977.

[8] Dennis, J.E. and Schnabel, R.B., *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, 1983.

[9] Garcia, C.B. and Zangwill, W.I., *Pathways to Solutions, Fixed Points, and Equilibria*, Prentice–Hall, Englewood Cliffs, 1981.

89-3276-CP

ADA IN A HARD REAL-TIME ENVIRONMENT

Bruce Jones
Harris Corporation, Computer Systems Division
Fort Lauderdale, Florida

Abstract

The Ada programming language is examined in the light of real-time scheduling theory. It is found that the current Ada standard makes the use of full Ada tasking in hard real-time systems problematic. A subset of the Ada tasking features can be used to implement a hard real-time system composed of independent Ada tasks. Such real-time systems can be analyzed under real-time scheduling theory if the tasks are assigned scheduling priorities according to the rate monotonic scheduling algorithm. The extension of scheduling theory to the full Ada tasking model and real-time systems composed of interdependent periodic tasks will require further research and modifications to the Ada language itself. Because these results will be slow to appear, the current Ada capabilities of the Harris Night Hawk family of real-time super-microcomputers are presented as an example of a short-term solution.

Introduction

This is a time of continuous and rapid advance in the computer technology available for real-time simulation and training systems. Commercially available microprocessors offer performance levels that meet or exceed the performance of traditional minicomputers. Multiprocessing computers offer improved price/performance ratios. Standard computer architectures, bus designs, and operating system interfaces allow efficient reuse of hardware and software components. The Department of Defense's (DoD) Ada mandate enforces a programming language standard while promising lower software development costs. The challenge for developers of real-time simulation and training systems is to integrate these new standards, requirements, and technology into their own development programs without affecting system performance or development cost.

While the effects of the development of standards and the new multi-microprocessor computer technology are to decrease system cost while increasing system performance, the benefits of the DoD's Ada mandate are not so obvious. Indeed, many workshops have been held and many papers have been published addressing the costs and deficiencies of Ada. The General Accounting Office's (GAO) recent report to the House Defense Appropriations Subcommittee concludes that "Defense has not yet demonstrated whether the use of Ada can help control software development and maintenance costs" and "... technical issues have affected the ability of Defense program managers to use Ada." [1] Although the technical issues observed by the GAO have yet to be resolved, the Ada mandate remains in force. It is assumed that the DoD will eventually demonstrate that the use of Ada can control and reduce software costs. The focus of this paper is the technical issues that affect the use of Ada tasking in real-time simulation and training systems.

Ada tasking is examined in the light of real-time scheduling theory. The results of that examination show that Ada tasking may be safely used in hard real-time systems if its use is restricted to a carefully chosen subset. Deficiencies are identified in the language that prevent the use of unrestricted Ada tasking in hard real-time systems.

Solving the problems of Ada tasking in a hard real-time environment requires changes to the language and continued research in real-time scheduling theory. Both are long-term solutions. The short-term solution is to avoid Ada tasking and construct Ada hard real-time systems using the traditional multiprogramming and executive approach.

Hard Real-time Systems

Real-time systems are defined as systems that react to external events in sufficient time to influence future events. Systems that are required to respond to events within absolute deadlines are termed "hard" real-time systems. Systems in which a statistical distribution of response times is acceptable are termed "soft" real-time systems.

A hard real-time system can be viewed as a collection of cooperating tasks, each responsible for handling one or more events. A system's tasks may be implemented as subroutines, Ada tasks, complete programs running under an operating system, or some combination of all three. A system's tasks may be independent or interdependent. Independent tasks do not communicate or synchronize with other tasks. Interdependent tasks may communicate or synchronize with, or even initiate, other tasks.

The events handled by a system's tasks may be aperiodic or periodic. To guarantee a given response time to periodic events, tasks can be scheduled to run periodically; for example, a task that is responsible for sampling an input stream at the rate of 20 times per second is coded to sample the input stream once, then is executed at the desired 20 Hz rate.

The phrase "frequency-based scheduling", or "FBS", denotes the periodic scheduling of a system's tasks. The term "deadline scheduling" has also been used to describe this scheduling technique. The executive responsible for implementing a frequency-based scheduling algorithm is referred to as a "frequency-based scheduler." This executive is sometimes referred to as a "cyclic executive."

The execution of a system composed of periodic tasks proceeds as a sequence of minor cycles. A real-time system's minor cycle is the smallest unit of frequency maintained by the system's scheduler; for example, a system composed of tasks running at 10 hz, 50 hz, and 100 hz has a minor cycle of 100 hz. Minor cycles are measured in terms of frequency or time: a 100 hz minor cycle is equivalent to a 10 millisecond minor cycle. It is common practice for periodic tasks to begin their execution on minor cycle boundaries and for their deadlines to occur on those same boundaries.

Performance and Schedulability

The requirements of a hard real-time system can be described in two terms: performance and schedulability. A task, either periodic or aperiodic, is said to meet its performance requirement if it is able to receive and react to external events within its deadline. A system (a collection of tasks) is said to meet its schedulability requirement if no combination of external events can cause any of its tasks to miss, or overrun, their deadlines. (Usage of the term schedulability follows that of [2] and [8].)

Task Performance Requirements

The performance of a given task is readily measured. As hardware and software performance levels improve over time, the cost of a given level of performance decreases. Although there will always be a need for faster hardware and more efficient software, technology has advanced to the point that the performance requirements of all but the most demanding Ada real-time systems can be satisfied with off-the-shelf hardware and software. Meeting a set of performance requirements is a question of evaluation and acquisition, both of which are outside the scope of this paper.

System Schedulability Requirements

The schedulability of a real-time system can affect its success. A system whose tasks consistently miss their deadlines will be viewed as unresponsive. Systems whose tasks overrun their deadlines during transient overload conditions will be viewed as unreliable and unpredictable. By definition, hard real-time systems cannot tolerate any deadline overruns.

Traditional methods that attempt to ensure schedulability, such as the maintenance of fifty percent spare processor capacity, do not guarantee schedulability. This method, and other such methods, can sometimes be replaced by a more rigorous analysis due to the development of real-time scheduling theory.

Real-time Scheduling Theory

Real-time scheduling theory is concerned with mathematically determining the schedulability of a real-time system. The theory has begun to produce important results. Its groundwork was laid by Liu and Layland in their landmark 1973 paper, which examined the rate monotonic scheduling algorithm [3]. The rate monotonic scheduling algorithm assigns tasks a scheduling priority based on their periods. Higher frequency tasks are assigned higher priorities than lower frequency tasks. Liu and Layland's paper provided two fundamental results: they proved that (1) for systems of independent tasks assigned static scheduling priorities, the rate monotonic algorithm is the optimal scheduling algorithm, and (2) the schedulability of such a system can be guaranteed by keeping the total processor utilization of the system's tasks less than approximately the natural logarithm of two, or 69 percent.

Liu and Layland's work, which shows that simple performance analysis can be used to determine the schedulability of limited classes of hard real-time systems, has recently been extended to more general classes of systems.

Sha, Rajkumar, and Lehoczky [2] at Carnegie Mellon University extended this work to include systems of interdependent periodic tasks running under the priority ceiling protocol. A system's interdependent tasks synchronize using only simple binary semaphores and communicate only through shared data objects protected by binary semaphores.

The priority ceiling protocol specifies the behavior of the semaphore operations. The priority ceiling protocol is an extension of the priority inheritance protocol. The priority inheritance protocol specifies that a low priority task which has locked a semaphore needed by a higher-priority task temporarily inherits the priority of the higher-priority task. The inheritance of the higher priority allows the low priority task to use and then release the resource protected by the semaphore. It is possible under the priority inheritance protocol for a high priority task to be blocked by several low priority tasks in succession. The priority ceiling protocol limits such blocking to the length of one critical section. The duration of each critical section in a program can be measured easily. The maximum blocking time for each task may then be determined by examining the duration of each critical section used by a given task. The schedulability of a system running under the priority ceiling protocol can then be determined by computing a formula involving each task's period, execution time, and maximum blocking time and comparing the result with Liu and Layland's processor utilization limit.

Lehoczky, Sha, and Strosnider [4] examined the schedulability of systems composed of a mixture of periodic and aperiodic tasks. They discussed the scheduling of a system's tasks under the deferrable server algorithm. This algorithm maximizes the response rate of aperiodic tasks without affecting the ability of the system's periodic tasks to meet their deadlines. They determined processor utilization bounds for a system's aperiodic tasks and for the system's total processor utilization that guarantees the system's schedulability.

The schedulability of a real-time system that uses Ada tasking is not so easily assured because real-time scheduling theory has yet to be extended to include the synchronization constructs of the full Ada tasking model. The use of a subset of Ada's tasking constructs will be shown to allow the application of real-time scheduling theory. Proving the schedulability of an Ada program that makes unrestricted use of Ada's tasking constructs may not be possible.

Ada in an Hard Real-time Environment

The Ada language contains a rich set of task synchronization and communication constructs and is well suited to constructing a system of aperiodic interdependent tasks. Hard real-time systems, however, often consist of a mixture of periodic and aperiodic tasks. There is no mechanism within the Ada language that directly supports periodic task scheduling. The Association for Computing Machinery's (ACM) SigAda Ada Runtime Environments Working Group (ARTEWG) has proposed a library package interface to a cyclic executive termed a "synchronous scheduler," [5] but this interface is not standard Ada and has not yet been implemented.

To use a cyclic executive to schedule Ada tasks, the executive itself must be coded in Ada, use the Ada tasking constructs, and be compiled as part of the real-time system.

Implementing an FBS Using Ada Tasking

There are several different techniques that may be used to implement an FBS system using Ada tasking. Two different approaches will be discussed. One is called the "self directed" model, and one is called the "executive" model. The analysis of these models will assume that the tasks are independent. The consequences of eliminating this assumption will be examined after both models are presented.

The Self Directed Model

In the self directed model, each task of the real-time system is coded as an Ada task. The individual tasks of the system schedule their own execution through the use of the Ada delay statement. Once initiated, each task enters a loop in which it first performs its minor cycle processing and then delays for a period of time calculated to end at the start of the task's next minor cycle. The task in Figure 1 illustrates the technique.

Several technical issues, as well as the example task's inherent complexity, prevent consideration of this model as an acceptable solution to the problem of implementing an FBS system using Ada tasks. These issues are the semantics of the Ada delay statement and the reliability of the timing computations.

The Ada delay statement is defined by the Ada reference manual (RM) in 9.6(1) [6] to "... suspend further execution of the task that executes the delay statement, for at least the duration specified ...." The approved language commentary AI-00032/09 [7] extends this definition by stating that the expiration of the delay executed by a high-priority task must cause the preemption of any running lower-priority tasks.

While the intent of AI-00032 is clearly to force implementations of predictable scheduling, the RM's use of the phrase "at least" hints at the delay statement's lack of determinism. The actual amount of time spent suspended at a delay depends heavily on the state of the other tasks in the system. The presence of higher or equal priority executing tasks will block the resumption of the suspended task. A task executing an Ada delay statement cannot make any assumptions as to when it will again resume execution.

Another technical issue that affects the reliability of the self directed model is the susceptibility of its timing calculations to corruption by the untimely arrival of interrupts. Processor time spent in handling interrupts in the time critical section between the actual minor cycle processing and the delay statement will cause the task to delay for a longer time than required and cause its deadline to be overrun. Similar and possibly larger overruns will arise on a multiprogramming system if the task loses its processor because of a context switch during that same section.

The execution of a system running under the self directed model is unpredictable because of the definition of the Ada delay statement and the model's inherent susceptibility to corruption. No determination of schedulability can be made for systems that use this model or the Ada delay statement.

The Executive Model

In the executive model, each task of the real-time system is coded as an Ada task. An additional task is created as the executive task. This task is responsible for starting the processing for each minor cycle and for detecting overruns. Each of the system's other tasks waits on an entry that the executive calls to indicate the start of its minor cycle processing. The task in Figure 2 illustrates this technique.

The executive task functions as a traditional cyclic executive:

```
task SAMPLE_INPUT is
   entry START;
end SAMPLE_INPUT;

with CALENDAR;
task body SAMPLE_INPUT is
   procedure PROCESS_INPUT is separate;
   procedure REPORT_OVERRUN is separate;
   PERIOD : constant DURATION := 0.10;
   START_TIME : CALENDAR.TIME;
   END_TIME   : CALENDAR.TIME;
   DELAY_TIME : DURATION;
begin
   accept START;
   loop
      START_TIME := CALENDAR.CLOCK;
      PROCESS_INPUT;
      END_TIME := CALENDAR.CLOCK;
      << START_OF_TIME_CRITICAL_SECTION >>
      if ( END_TIME > START_TIME + PERIOD ) then
         REPORT_OVERRUN;
         while ( END_TIME > START_TIME + PERIOD ) loop
            END_TIME := END_TIME - PERIOD;
         end loop;
      end if;
      DELAY_TIME := PERIOD - ( END_TIME - START_TIME );
      << END_OF_TIME_CRITICAL_SECTION >>
      delay DELAY_TIME;
   end loop;
end SAMPLE_INPUT;
```

Figure 1. Example FBS Task Under the Self Directed Mode!

it uses conditional entry calls both to initiate each task's minor cycle and to detect overruns. Portions of a sample executive task are shown in Figure 3.

```
task SAMPLE_INPUT is
   entry START_MINOR_CYCLE;
end SAMPLE_INPUT;

task body SAMPLE_INPUT is
   procedure PROCESS_INPUT
      is separate;
begin
   loop
      accept START_MINOR_CYCLE;
      PROCESS_INPUT;
   end loop;
end SAMPLE_INPUT;
```

Figure 2. Example Executive Model FBS Task

The executive task attaches itself to an interrupt, such as a real-time clock interrupt. The occurrence of the interrupt marks the start of a minor cycle. During the rendezvous with the interrupt, the executive task performs an entry call to each task that should run during the minor cycle. Tasks that accept the call start their minor cycle execution; a failure to accept the call is recorded by the executive as an overrun.

The simplicity of the tasks that run under the executive model is attractive, and the model does not suffer from the problems of the self directed model. Tasks scheduled under the self directed model attempt to control the duration of their idle time but fail in the attempt. The executive model instead relies on the executive task to control the beginning of each task's minor cycle. The troublesome semantics of the delay statement are avoided, and the use of a hardware interrupt frees the system from the need to perform timing calculations.

A system of independent periodic Ada tasks that is prioritized under the rate monotonic scheduling algorithm and running under the executive model can be analyzed under real-time scheduling theory. The executive task is treated as a traditional cyclic executive; the rendezvous is considered equivalent to a cyclic executive's "wakeup" call; and the overhead of performing the rendezvous is added to each task's processor utilization. Liu and Layland's formula [3] may be used to de-

```
task EXECUTIVE is
   entry INTR;
   for INTR use at 16#01#;
end;

task body EXECUTIVE is
   CYCLE : INTEGER := 0;
begin
   loop
      accept INTR do
         CYCLE := ( CYCLE rem NUMBER_MINOR_CYCLES ) + 1;
         if ( RUN_THIS_CYCLE( SAMPLE_INPUT_TASK, CYCLE ) ) then
            begin
               select
                  SAMPLE_INPUT.START_MINOR_CYCLE;
               else
                  REPORT_OVERRUN( SAMPLE_INPUT_TASK );
               end select;
            exception
               when TASKING_ERROR =>
                  REPORT_FAILURE( SAMPLE_INPUT_TASK ):
            end;
         end if;
      end INTR;
   end loop;
end EXECUTIVE;
```

Figure 3. Ada FBS Executive Task Code Fragments

termine the system's schedulability.

Interdependent Ada Tasks

The analysis of the executive model under real-time scneduling theory is based on the assumptions that: (1) all tasks are independent, and (2) all rendezvous are limited to the simple "wakeup" rendezvous used by the executive. Both assumptions must be set aside in order to analyze systems of interdependent Ada tasks. This analysis must be performed in the light of the implicit priority inversions that can arise through the unrestricted use of Ada rendezvous.

Priority Inversions in Ada Tasking

A priority inversion is said to occur in a multitask application when a lower priority task blocks the execution of a higher priority task. Priority inversions can never be completely prevented because low priority tasks must be allowed to execute within critical sections. Priority inversions can significantly affect the reliability and responsiveness of a real-time system. Unbounded priority inversions prevent a rigorous schedulability analysis under the current theory.

Three sources of implicit priority inversion were identifieo within the Ada language by the General Accounting Office's report to the House: the queuing of entry calls, the priority semantics of a rendezvous, and the use of tasks as interrupt handlers. Only the first two of these three issues will be discussed because interrupt handling tasks are outside the scope of this paper. A fourth source of priority inversion that was not mentioned by the GAO, the Ada selective wait statement, will also be discussed.

Queuing Priority Inversion

Assume that a real-time system is composed of three Ada tasks: T1, T2, and T3. Let the priorities of the three tasks be ordered such that T1 has the highest priority, T2 has an intermediate priority, and T3 has the lowest priority. Assume that all three tasks are to be executed on the same physical processor. If T1 and T2 both request a rendezvous with T3 and T2's request precedes T1's request, the higher-priority task, T1, will be blocked until the rendezvous between T2 and T3 finishes.

The language's requirement that task entry calls must be queued in arrival, or first in first out (FIFO), order is the direct source of this inversion. The inversion can be removed by changing the language to allow priority sorting of entry queues, or it can be avoided as suggested by Sha and Goodenough through careful design and use of entry families [8]. Their suggestion removes the inversion by preventing entry queues from forming. The suggested solution's cumbersome nature shows the need for a change to the language that allows priority sorted entry queues.

Rendezvous Priority Inversion

Using the same set of Ada tasks - T1, T2, and T3 - assume that T1 requests a rendezvous with T3 and that T3 is not ready to accept the call. T1 must wait, and according to RM 9.8(4), the Ada runtime system is required to run T2 if it is runnable, therefore preventing T3 from accepting T1's call and thus blocking T1.

The source of this inversion is a strict interpretation of RM 9.8(4), which states that "If two tasks with different priorities are both eligible for execution and could sensibly be executed using the same physical processors ... then it cannot be the case that the task with the lower priority is executing while the task with the higher priority is not." If it is assumed that the RM's use of the word "sensible" can allow the implementation of a priority inheritance protocol, the inversion can be avoided. The recent release of the language commentary AI-00594/02 by the Ada Raporteur Group attempts to address this issue; it specifically discusses the priority inheritance protocol as an example of a "rational global policy" for task scheduling that is considered "sensible" even though it appears to be in direct conflict with the language's requirements for preemptive scheduling.

Unfortunately for the Ada community, AI-00594 is quite controversial and is opposed by the ARTEWG. While there are significant problems with Ada task scheduling, the freedom AI-00594 gives implementations to define their own "sensible" scheduling protocols will seriously affect the portability of Ada programs. This issue cannot be considered fully resolved.

Selective Wait Priority Inversion

Again, using the same set of Ada tasks - T1, T2, and T3 - assume that T3 has two entries E1 and E2, and assume that T3 executes a selective wait including both entries among the open alternatives. Let T1 and T2 perform entry calls to E1 and E2, respectively. The current definition of the language does not prevent T3 from accepting the rendezvous with T2 first, although this denies service to the higher priority T1.

The Ada standard explicitly allows this inversion, for it states in RM 9.7.1(6) that if " ... several alternatives can thus be selected, one of them is selected arbitrarily (that is, the language does not define which one)." While it is hoped that implementations of Ada tasking systems will choose from the open alternatives by priority, the language does not enforce such a choice. The wording of the standard should be changed to specify that the choice must be made on the basis of priorities.

The Effect of Priority Inversion on Schedulability Analysis

The determination of the schedulability of a system composed of independent periodic tasks depends upon two factors: the period of each task T(*t*) and the execution time of each task C(*t*). The extension of this theory to systems composed of interdependent tasks introduces another factor B(*t*), the amount of time a given task is prevented from executing by the execution of a lower priority task within a critical section. A rigorous analysis is only possible under the priority ceiling protocol because that protocol limits the length of time a task may be blocked to the length of one critical section. The factor B(*t*) represents the worst case blocking time for each task, or the longest time spent by other tasks in any of the critical sections used by task *t*.

The unbounded priority inversions implicit in Ada tasking prevent the determination of the length of time a task may be blocked, and therefore prevent a rigorous schedulability analysis.

Proposed Ada Language Changes

The following list summarizes the proposed changes to the Ada standard.

- Augment the language with a deterministic delay construct, perhaps similar to the ARTEWG's proposed BOUNDED_DELAYS package [5].
- Remove the requirement of FIFO sorting for entry queues, and require the sorting of entry queues by task priority.
- Require the use of task priorities in choosing from the open alternatives of a selective wait statement.
- Allow a "sensible" rendezvous implementation to use the priority inheritance protocol.
- Provide a standard method of scheduling tasks in a frequency-based manner, such as the ARTEWG's synchronous scheduling package.

Research Directions

The language changes proposed, if adopted as part of a future Ada standard, will provide solutions to three important problems of Ada in a hard real-time environment. They provide for: (1) the deterministic control of time, (2) the elimination of implicit priority inversions, and (3) the periodic scheduling of tasks. These language changes are necessary for Ada tasking to reliably support complex and interdependent hard real-time systems. They are not sufficient to allow the application of real-time scheduling theory to full Ada tasking.

Real-time scheduling theory must be extended to include Ada tasking constructs, or Ada tasking constructs must be implemented as the binary semaphore operations already covered by the current theory. Both of these areas are subjects for further research.

Short-term Solutions

Proposing language changes is a long-term solution to the problems of Ada in a hard real-time environment. Awaiting the results of further research in real-time scheduling theory is also a long-term solution.

The Software Engineering Institute (SEI) suggests in its Ada Adoption Handbook [9] that the "... near term answer is simply to avoid using Ada tasks for systems that have critical real-time requirements." The analysis of the executive model under real-time scheduling theory shows that this position is overly pessimistic; however, for most hard real-time systems, the SEI's suggested solution of continued reliance on cyclic executives is appropriate.

The Harris Night Hawk

The Harris Night Hawk family of multiprocessing supermicrocomputers is an example of an off-the-shelf implementation of a near-term solution. The Night Hawk integrates standard hardware components, an efficient real-time cyclic executive, and a highly optimizing Ada compiler into a high performance multiprocessor platform that is ideally suited to real-time simulation and training systems.

Architecture

The current generation of the Night Hawk family is based on the Motorola 68030 micro-processor. A Night Hawk system may be configured with one processor board as a single processor system or with up to eight processor boards to create a multiprocessor system.

All of the processor boards in a Night Hawk system share access to a common pool of global memory. Each processor also has its own on-board local memory to reduce contention for the system bus. Contention is further reduced by the large global memory cache built into each processor board.

The Night Hawk supports both the VME and SCSI bus standards for such peripheral devices as discs, tape drives, and network controllers. The MIL-STD 1553B, Gould HSD, and Digital Equipment DR11 bus standards are also supported to allow Night Hawk systems to interface with specialized real-time devices.

Real-time Executive

The Night Hawk supports three different but compatible operating systems. All are multi-threaded and fully support multiprocessing. The CX/UX operating system is a general purpose UNIX [10] operating system compatible with both BSD4.2 (Berkeley Standard Distribution) and AT&T System V.2. The CX/SX operating system is a secure version of the same operating system and is currently undergoing B1 level security evaluation. The CX/RT operating system is the Night Hawk's real-time operating system.

The Night Hawk's CX/RT operating system has been optimized for fast execution speed, interrupt response, and context switching. In addition to the normal services of an operating system, it provides:

- An efficient frequency-based scheduler (FBS) to allow programs to be scheduled in a periodic manner.
- A high resolution performance monitoring subsystem to capture the execution time of programs scheduled under an FBS to microsecond levels.
- A real-time and interactive data recording subsystem to allow the examination and modification of program data objects.

Ada Compiler

The Night Hawk's Ada compiler is based on the Verdix [11] Ada Development System (VADS) [12]. The Verdix code generator has been replaced with Harris' proprietary Common Code Generator, and the development environment has been augmented with a wide range of software engineering tools. The Night Hawk Ada compiler has been validated under the Ada Compiler Validation Capability (ACVC) version 1.10.

The Night Hawk Ada compiler's common optimizer performs a vast array of global optimizations, among which are dead or unreachable code elimination; comprehensive strength reduction; constant, variable and expression propagation; common subexpression identification; inline expansion of subprograms; folding of partially constant boolean expressions; and loop invariant code motion. It performs several optimizations specific to the Ada language, such as constraint propagation and the replacement of raised exceptions with direct branches to the appropriate exception handler.

The Night Hawk Ada compiler provides support for the sharing of data objects between programs. Shared memory is the fastest and most efficient method of implementing inter-program communication. Data objects may be placed in shared memory through the use of the operating system's shared memory services or through mechanisms built into the compiler.

The Night Hawk Ada compiler supports the implementation-defined pragma "shared_package." All data objects declared within a library level package marked with pragma shared_package will be allocated within shared memory regions and will be shared by each program that references the package. To allow programs to control access to this shared data, the compiler provides the implementation-defined attributes "'LOCK" and "'UNLOCK", which provide binary semaphore operations for shared packages.

FBS Systems on the Night Hawk

Programs scheduled under the Night Hawk's FBS are allowed to execute until they call the system's FBS_Wait service. Programs that call FBS_Wait are suspended until the beginning of their next scheduled minor cycle and then are released and allowed to continue execution. Figure 4 presents an example of an Ada program that can be scheduled under the Night Hawk's FBS executive.

```
with RT_INTERFACE;
procedure SAMPLE_INPUT is
   ISTAT : INTEGER;
   procedure PROCESS_INPUT is separate;
begin
   loop
      RT_INTERFACE.FBS_WAIT( ISTAT );
      exit when ISTAT /= 0;
      PROCESS_INPUT;
   end loop;
end SAMPLE_INPUT;
```

Figure 4. Example Night Hawk FBS Program

Scheduling Theory and the Night Hawk

Real-time systems (collections of programs) running under the CX/RT executive may be assigned scheduling priorities based on the rate monotonic scheduling algorithm. Real-time scheduling theory may be applied to such systems if the programs are independent.

Scheduling theory cannot currently be applied to systems composed of interdependent tasks running under CX/RT, for while the operating system does provide binary semaphore services, the semaphores do not currently implement priority inheritance or the priority ceiling protocol. The semaphore operations, as currently implemented, have been optimized for execution speed.

It is possible to create a system composed of tasks that share access to common data that behaves as an independent system. Tasks that share data can simply be scheduled so that they never run during the same minor cycle. The system then relies on the cyclic executive to ensure the integrity of the shared data; task independence is preserved; and real-time scheduling theory may be applied.

Night Hawk Benchmark Results

A review of the Night Hawk's capabilities is not complete without the discussion of benchmark results. Unless otherwise noted, all of the results below have been obtained from Ada programs compiled under optimization and with runtime checks suppressed. The programs have been executed on a Night Hawk HN3808 computer system.

Processor Performance

There are several traditional benchmarks that measure the execution speed of a processor. Figure 5 shows the Night Hawk's results for these benchmarks.

Whetstone MIPS per second	6.06
Dhrystone Instructions per second	5258
Milliseconds for one Dhrystone instruction	0.1901

Figure 5. Night Hawk Benchmark Results

The ACM SigAda's Performance Issues Working Group (PIWG) has published a suite of Ada benchmarks. A small subset of the Night Hawk's results on these benchmarks is presented in Figure 6. All results are in microseconds.

T000001	114.6	T000005	118.8
T000002	114.6	T000006	191.6
T000003	117.2	T000007	114.6
T000004	151.0	T000008	322.9

Figure 6. Night Hawk PIWG Results

The results of the PIWG rendezvous benchmarks demonstrate the feasibility of the executive model. The rendezvous time of approximately 120 microseconds is quite small compared to the 10,000 microsecond length of a 100 hz minor cycle. In fact, this performance is only slightly slower than the overhead of the Night Hawk's frequency-based scheduler.

Multiprocessor Contention

The Night Hawk's large global memory caches significantly reduce contention for the system's bus. Small benchmarks, such as the Whetstone benchmark, can run entirely within the cache. Eight copies of Whetstone running in parallel on an eight proc-

CPUs	Simulation Execution Time in Seconds	Best / Worst	% Increase
1	243	243	0.0
2	245 245	245	0.82
3	247 246 247	246 / 247	1.23 / 1.64
4	248 249 249 249	248 / 249	2.05 / 2.46
5	250 251 251 250 250	250 / 251	2.88 / 3.29
6	254 253 254 254 257 256	253 / 257	4.11 / 5.76
7	258 259 259 260 264 260 263	258 / 264	6.17 / 8.64
8	264 261 267 267 260 268 270 273	260 / 273	6.99 / 12.34

Figure 7. Night Hawk Multiprocessor Contention

essor Night Hawk execute in the time taken by one copy running on one processor.

Programs may be loaded into a processor's on-board local memory and executed in parallel with other programs on other processors with no degradation in performance.

Because the amount of local memory on each processor board is limited, some portions of a real-time system may run in global memory. If the programs in global memory are large enough to cause cache misses, contention for the system's memory bus will affect multiprocessor performance. The table shown in Figure 7 shows the results of executing a large helicopter simulation on various Night Hawk HN3800 configurations. In each case, one processor is dedicated to running one copy of the simulation. The worst case degradation due to bus contention is shown to be about twelve percent.

Cyclic Executive Overhead

The overhead of a cyclic executive can be measured in several ways, but perhaps the easiest is to measure the performance of

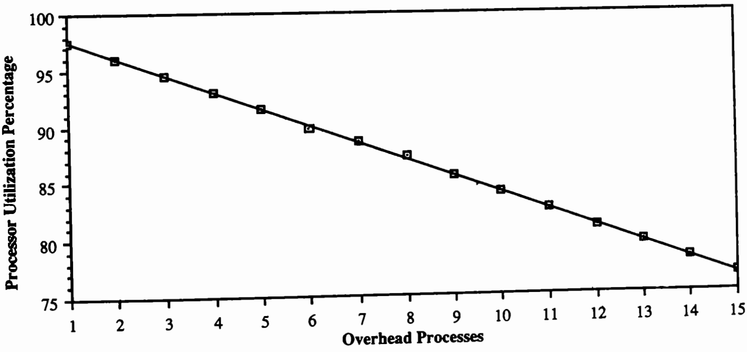


Figure 8. Night Hawk Cyclic Executive Efficiency

tasks running under the executive. Imagine a system composed of one task. The amount of processor time available to that task during a minor cycle, when compared with the duration of the minor cycle itself, can be used to determine the executive's overhead.

The addition of other tasks to the system will increase the executive's overhead. If these additional tasks are limited to requesting immediate context switches, and run at a higher priority than the "workload" task, the executive's context switching overhead can be determined by measuring the amount of processor time available to the workload task.

Figure 8 shows the amount of processor time available to the workload task when various numbers of overhead tasks are scheduled. The graph demonstrates the high efficiency of the CX/RT executive and shows the gradual increase in overhead due to an increase in the number of scheduled programs

Conclusions

The Ada programming language promises many benefits, including the ability to program complex real-time systems in a standard, portable language. This examination of Ada's suitability for use in hard real-time systems has shown that while the schedulability, and therefore the reliability, of systems composed of independent Ada tasks limited to a subset of Ada's tasking features can be determined, the analysis cannot be extended to the more general case of interdependent tasks. Given the complexity and interdependence of most real-time systems, it is expected that the promise of Ada will remain unfulfilled until the language itself is changed to address these issues.

Although the problems of Ada tasking in a hard real-time environment are severe, they do not prevent the development of successful simulation and training systems. Ada tasking cannot be used to its fullest extent, but even its use when limited to independent tasks may provide some benefits. The other benefits of the language, such as its support for reusable code and data encapsulation, can still be obtained even if tasking is avoided entirely. The traditional methods of constructing hard real-time systems, such as the use of cyclic executives, are still available.

Acknowledgements

I would like to extend thanks to a long list of my co-workers, including Rajiv Sinha, Jeff Hollensen, Paul Earhart, Carol Charlson, Anne Thayer, Ralph Capasso, Marge Shallenberger, Bret Needle, Anne McMichael, Joel Horn, Mike Hohulin, and Mark Heuser for their time, advice, encouragement, and support, and for proofreading this paper. I would also like to thank Brian Leach for his help in preparing the manuscript for publication.

Notes

[1] General Accounting Office
Defense's Implementation of Ada
GAO Report IMTEC-89-9, March 1989

[2] Lui Sha, Ragunathan Rajkumar, and John P. Lehoczky
Priority Inheritance Protocols
Carnegie-Mellon University technical report, CMU-CS-87-181, November 1987

[3] C. L. Liu and James W. Layland
Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment
Journal of the ACM, Vol. 20, No. 1, January 1973

[4] J. P. Lehoczky, L. Sha, J. Strosnider
Enhancing Aperiodic Responsiveness in A Hard Real-Time Environment
Proceedings of the IEEE Real-Time Systems Symposium, 1987

[5] ACM SigAda Runtime Environments Working Group
A Catalog of Interface Features and Options for the Ada Runtime Environment
ARTEWG, December 1987, p 3-24

[6] United States Department of Defense
Reference Manual for the Ada Programming Language
ANSI/MIL-STD-1815A-1983, February, 1983

[7] Grebyn Corporation
The Approved Ada Language Commentaries
Grebyn Corporation, 1988

[8] Lui Sha and John B. Goodenough
Real-Time Scheduling Theory and Ada
Carnegie-Mellon University technical report, CMU/SEI-88-TR-33, 1988

[9] John Foreman and John Goodenough
Ada Adoption Handbook: A Program Managers Guide
Carnegie-Mellon University technical report, CMU/SEI-87-TR-9

[10] UNIX is a registered trademark of AT&T.

[11] Verdix is a trademark of Verdix Corporation.

[12] VADS is a trademark of Verdix Corporation.

THE SEARCH FOR A NEW FLIGHT SIMULATION SOFTWARE MODELING STANDARD: A MODULAR APPROACH USING FEATURES OF THE ADA PROGRAMMING LANGUAGE

89-3278-CP

Glen Glasell

Karl Forsstrom*

Northrop Corporation
B-2 Division
Pico Rivera, CA

ABSTRACT

The Northrop B-2 Division Flight Simulation Laboratory has developed a real time flight simulation program written entirely in MIL-STD 1815A Ada Programming Language. The goal of the project was to develop a common flight simulation environment that can be used to support Operational Flight Program (OFP) development written in Ada. With this capability in place on a general purpose flight simulator (computer, cockpit, visual, motion system, etc.), the development, modification and verification of the real aircraft OFPs will be greatly enhanced.

BACKGROUND

Real time flight simulation has always struggled with developing a universal simulation model and with balancing the fidelity requirements of that model with computational resources. Within the flight simulation community, there has, in the past, been little standardization of the basic math models. Even fundamental concepts such as coordinate systems and aircraft parameter naming conventions were independently developed and promulgated by numerous companies and institutions. Requirements for implementing those increasingly complex math models of the vehicle dynamics and avionics systems generated demands for ever increasing computational power and flexibility in both hardware and software.

As the computer hardware improved, the programming methods evolved from analog computer patch panels to higher order programming languages (HOL) on digital computers. The computational inefficiencies of higher order languages such as Fortran were more than offset by the increased speed of the computers and ease of programming HOLs compared to coding in assembly language. The recent introduction of the Ada Programming Language is the latest step in this evolution towards more sophisticated programming as well as towards a Department of Defense directed standardization of computer programming.

The Air Force's Modular Simulation (ModSim) Project (now known as "Have Module") is an example of a standardization effort in flight simulation, but it is mainly concerned with creating a standard for <u>interfacing</u> simulation modules such as aerodynamics, propulsion, avionics, etc., rather than the detailed <u>development</u> of math models and the <u>programming</u> of those models. The time seems ripe to meld a standard simulation math model with a standard programming language and to optimize the best features of the model with those of the language.

<u>Northrop IR&D Simulation Projects</u>

Northrop Corporation, B-2 Division, has over the last five years been engaged in Independent Research & Development (IR&D) projects to develop new flight simulation technology. One IR&D project developed a generic rotating, round earth equations-of-motion model written, in large part, using vector notation. The model is capable of accurate round the world flight and is flexible enough to accomodate any conventional aircraft.

Use of vector notation can be of great use and convenience in programming mathematical models of dynamic systems. Preparation and groundwork required is essentially that of defining the reference systems, triads and rotation sequences to be used for each dynamic system used.

While this initial preparation may be tedious and addictive to detail, benefits accrue as the model is developed and expanded to include additional dynamic systems. As long as consistency with basic definitions is maintained, addition of dynamics systems to the basic model becomes straightforward. The effort associated with the details of determining specific signum characteristics of sub-components becomes almost automated within the mechanics of matrix algebra. Thus, the sometime bewildering dynamics coordinate concepts relating to "up", "down", "left" and "right" simply fall out of the vector/matrix equations. The designer's efforts can be directed towards the fundamental equations and nature of transformations rather than towards computational mechanics.

The classical rotational and linear equations of motion can be used as illustrations of this vector approach to modeling. The linear and angular vector momentum equations in a rotating frame are solved for the linear and angular vector rates:

$$\underline{\dot{V}} = (1/m)\ \{\mathrm{Sum}(\underline{F}) - (\underline{\omega} \times \underline{V})\}$$

$$\underline{\dot{\omega}} = [I]^{-1}\{\mathrm{Sum}(\underline{M}) - (\underline{\omega} \times \underline{I\omega})\}$$

where $\underline{\omega} = [p,q,r]^{-1}$, rotational rate vector
$\underline{V} = [u,v,w]^{-1}$, linear rate vector
$\underline{M} = [\mathcal{L},\mathcal{M},\mathcal{N}]^{-1}$, moments vector
$\underline{F} = [X,Y,Z]^{-1}$, force vector
$[I]$ = Inertia tensor
m = mass

The vector approach was used extensively throughout the development of the entire dynamics model starting with the spherical earth coordinate system (longitude, latitude, earth radius) transformed to a tangent plane local cartesian coordinate system (x,y,z) and including any inertial to body axes transformations and ground reactions or gear dynamics. Quaternions can be included in the modeling by creating special operators which behave like and are usable with vector operations.

*Member AIAA

A Fortran version of this new math model has been verified with existing Fortran models and is documented in Northrop IR&D Annual Plan 85-D-1012 (Reference 1).

The goal of a second IR&D project was to implement this round earth math model using the Ada Programming Language. Ada was chosen for several reasons. The primary reason was to comply with the DoD directives for the use of Ada software in all future mission critical embedded processors and the laboratory programs developed to support the design, analysis and simulation of the Operational Flight Programs (OFP) in those embedded processors. A secondary reason was to integrate Ada's unique programming features such as strong typing, array handling and ability to overload operators in the highly vectorized and modularized math model developed from the first IR&D project.

The strong typing features of Ada enable the programmer to identify variables as scalars, vectors and matrices. The Ada feature known as "overloading of operators" permits the programmer to redefine names or symbols to perform different functions based on the "type" of variable being operated upon. Hence, the Ada statement "a:= b * c" may define a scalar, vector or matrix multiplication depending on the prescribed "types" of variables a, b and c.

To implement this vector/matrix oriented model in Ada, an extensive Vector and Matrix Library Ada Package was written to enhance the programming performance and source code legibility. Northrop also developed and verified a Control Utility Ada Package containing typical dynamic modules such as table look-ups, filters (lags, leads, etc.) and integrators in order to simplify flight control law software programming.

The end result of this marriage between a vector oriented math model and the Ada Programming language is a simulation program of exceptional "built-in" documentation and efficient, modular code. A few examples will illustrate the style of programming.

The Ada Package vector_math defines numerous vector/matrix operations including overloaded operators (functions) for cross products, dot products of vectors and various combinations of scalar, vector and matrix summation, subtraction and multiplication. It also defines overloaded operators for matrix transposition and inversion. An abbreviated listing of package vector_math is shown below.

```
package vector_math is

   type vector is array(1..3) of float;
   type matrix is array(1..3,1..3) of float;
   procedure xprod(a,b: in vector; c: out vector);
   function xprod(a,b:vector) return vector;
   function "*"(b:vector; m:matrix) return vector;
   function "*"(m:matrix; b:vector) return vector;
   function "*"(s:float; m:matrix) return matrix;
   function "*"(s:float; v:vector) return vector;
   function "*"(a:matrix; b:matrix) return matrix;
   function "+"(a:vector; b:vector) return vector;
   function "-"(a:vector; b:vector) return vector;
   function transpose_matrix(m: matrix) return
            matrix;
   function inv_matrix(m: matrix) return matrix;

end vector_math;
```

The following are examples of the Ada code for some of the above functions:

```
package body vector_math is
   ----------------------------------------------------
--| function computes the x-product of 2 vectors.|
   ----------------------------------------------------
   function xprod(a,b:vector) return vector is
   c: vector;
   begin
      c(1) := a(2) * b(3) - a(3) * b(2);
      c(2) := a(3) * b(1) - a(1) * b(3);
      c(3) := a(1) * b(2) - a(2) * b(1);
      return c;
   end xprod;



   ----------------------------------------------------
--| Function multiplies 3x3 matrix by a 3x1        |
  | vector.                                        |
   ----------------------------------------------------
   function "*"(b:vector; m:matrix) return vector
            is
   v:vector;
   begin
      v(1) := b(1) * m(1,1) + b(2) * m(1,2) + b(3)
              * m(1,3);
      v(2) := b(1) * m(2,1) + b(2) * m(2,2) + b(3)
              * m(2,3);
      v(3) := b(1) * m(3,1) + b(2) * m(3,2) + b(3)
              * m(3,3);
      return v;
   end "*";

   -------------------------------------------
--| Function multiplies two 3x3 matrices |
   -------------------------------------------
   function "*"(a:matrix; b:matrix) return matrix
            is
      c: matrix;
   begin
      c(1,1) := a(1,1) * b(1,1) + a(1,2) * b(2,1)
                + a(1,3) * b(3,1);
      c(1,2) := a(1,1) * b(1,2) + a(1,2) * b(2,2)
                + a(1,3) * b(3,2);
      c(1,3) := a(1,1) * b(1,3) + a(1,2) * b(2,3)
                + a(1,3) * b(3,3);

      c(2,1) := a(2,1) * b(1,1) + a(2,2) * b(2,1)
                + a(2,3) * b(3,1);
      c(2,2) := a(2,1) * b(1,2) + a(2,2) * b(2,2)
                + a(2,3) * b(3,2);
      c(2,3) := a(2,1) * b(1,3) + a(2,2) * b(2,3)
                + a(2,3) * b(3,3);

      c(3,1) := a(3,1) * b(1,1) + a(3,2) * b(2,1)
                + a(3,3) * b(3,1);
      c(3,2) := a(3,1) * b(1,2) + a(3,2) * b(2,2)
                + a(3,3) * b(3,2);
      c(3,3) := a(3,1) * b(1,3) + a(3,2) * b(2,3)
                + a(3,3) * b(3,3);
      return c;
   end "*";
```

To illustrate the use of these overloaded operators, let's look at a very small section of package round_earth_eom, the round earth equations-of-motion model. One of the final steps in that model is the computation of the aircraft linear and angular accelerations. The accelerations are defined as two vectors, each containing three elements. The linear acceleration vector is called v_dot_v_b and the angular acceleration vector is called omega_dot_v_b. The suffix "v" identifying the variable as a type vector and the suffix "b" identifies the reference frame as body.

The actual Ada code closely resembles the "symbolic" vector notation that was illustrated above. For each operation, the program "knows" when to interpret the "*" symbol as a scalar times a vector operation, a vector times a vector operation or a matrix times a matrix operation.

```
   procedure compute_acc is
   ---------------------------------------------
--| description:                                |
--|    procedure computes linear and angular    |
  |    accelerations from the forces            |
--|    & moments.                               |
--|                                             |
--| package inputs & outputs -                  |
--|                                             |
--| aircraft:                                   |
--|    objects        mode  type   description  |
--|    mass           in    float  mass         |
--|    inertia_t      in    matrix inertia tensor|
--|                                             |
--| eom:                                        |
--|    objects        mode  type   description  |
--|    sum_force_v_b  in    vector sum of the   |
--|                                forces       |
--|    sum_torque_v_b in    vector sum of the   |
--|                                torques      |
--|    v_dot_v_b      out   vector linear acc.  |
--|    omega_dot_v_b  out   vector angular acc. |
--|                                             |
   ---------------------------------------------

   begin



   ------------------------------------------
--| Compute linear accelerations in body axis |
   ------------------------------------------
      v_dot_v_b := (1.0/mass) * (sum_force_v_b -
                   xprod(omega_v_b,velocity_v_b))

   -------------------------------------------
--| Compute angular accelerations in body axis |
   -------------------------------------------
      omega_dot_v_b := inv_matrix(inertia_t) *
                       (sum_torque_v_b -
                       xprod(omega_v_b, inertia_t
                            * omega_v_b));

   end compute_acc;
```

The resultant Ada code closely parallels the symbolic representation and the strong typing of the variables ensures that like variable types are used for each mathematical operation. Similar simplification and legibility of code can be seen in other simulation modules. The flight control Ada package for the IR&D simulation is much cleaner and straight-forward than the equivalent Fortran subroutine.

Conclusion

The full Ada implementation of the simulation is documented in Northrop IR&D Annual Plan 89-D-8075 (Reference 2) and CTDC Annual Plan 89-C-8370 (Reference 3). Several portions of the CTDC report are classified Secret. Subsequently, an unclassified F-5E simulation has been implemented and is currently being used by Northrop and by California Polytechnical State University in San Luis Obispo to investigate advanced simulation methods including incorporation of MIL-STD 1553 communications protocols and other newer communications systems.

As mentioned earlier, one of the intents of this project was to enhance the design, integration and test of Operational Flight Programs, the embedded software used in aircraft and other weapon systems for which the Ada language was originally designed. Northrop's engineers have enhanced the simulation laboratory's capabilities by developing the necessary math modeling and programming techniques needed to reach this goal.

List of References

1. Piloted Simulation Model Improvement, IR&D Project 86-D-1012 Annual Plan (Proprietary), Northrop Corporation, B-2 Division, Dec., 1985

2. Ada Software Development, IR&D Project 89-D-8075 Annual Plan (Proprietary), Northrop Corporation, B-2 Division, Dec., 1988

3. Simulator Software Checkout and Documentation, CTDC Project 89-C-8340 Annual Plan (Proprietary), Northrop Corporation, B-2 Division, Dec., 1988

A Comparison of the Ground Effects Measured With and Without Recent-of-Descent Modeling on the F-15 S/MTD Configuration

Guy T. Kemmerly
NASA Langley Research Center
Hampton, VA 23665-5225

Abstract

A moving-model ground-effects testing method has been used to study the influence of rate-of-descent on the approach aerodynamics of the F-15 S/MTD configuration. A comparison of the results obtained while modeling rate of descent to data obtained in a wind tunnel with zero rate of descent showed significant differences due both to the rate of descent in the moving-model test and to the presence of the ground boundary layer in the wind-tunnel test. Relative to the conventional static wind tunnel ground-effects tests, the rate-of-descent modeling produced substantially less lift increase in ground effect, less nose-down pitching moment, and less increase in drag. These differences became more prominent at the larger thrust vector angles.

The results of this investigation indicate no safety-of-flight problems with the lower reverser vectored up to 80° on approach. They also indicate that this configuration could employ a nozzle concept using lower reverser vector angles up to 110° on approach if an approach procedure were adopted in which rate of descent was not arrested near the ground and if inlet reingestion were found not to pose a problem.

Introduction

Quite often, in the history of aviation, the ground effects measured during flight tests of some aircraft on approach have not matched those predicted based on steady state wind tunnel test results. Some examples of this can be found described in references 1, 2, and 3. The configurations that fall into this category are generally low aspect ratio and/or configurations employing some form of vectored thrust. However, when the ground effects were measured on a low aspect ratio aircraft flown at constant altitudes near the ground, it was found that the measurements matched well with those predicted in the wind tunnel test (reference 4). This suggests that conventional wind tunnel ground-effects tests (i.e. time-averaged tests of a stationary model at various ground heights) actually simulate an aircraft flying near the ground at a particular altitude rather than an aircraft descending through that altitude as is the case in an actual aircraft approach. In fact, it was found that, though the ground effects measured in the flight tests of the XB-70 aircraft did not match those predicted by the static wind tunnel tests of the configuration, when rate-of-descent was simulated in the testing a much better match was obtained. This is illustrated in figure 1 and reported in reference 5.

As a result of these findings, the NASA Langley Research Center developed the capability to measure the ground effects on powered and unpowered models subjected to a constant rate of descent using the Vortex Research Facility (VRF) at the NASA LaRC. Several models have been tested using this capability and then tested again conventionally in the 14- by 22-Foot Subsonic Tunnel also at Langley (reference 6). In comparing the results one conclusion is clear: rate-of-descent modeling has a significant impact on the measured ground effects of a configuration.

Figure 2 illustrates some of the important differences between conventional static ground effects test methods and the moving model method. Static test techniques involve setting a model at a given height above the ground plane, allowing the flow field to reach a steady state, and measuring the aerodynamic loads. The moving-model technique, on the other hand, involves measuring the aerodynamics while the model is in motion and the flowfield is in a dynamic state, similar to conditions in an actual approach. Simulations of normal approaches (without thrust reversers) have indicated only small, but discernable, differences in model aerodynamics measured statically and at various rates of descent. With thrust reversers or similar jet devices operating, however, the two techniques could yield significantly different results. There are two primary reasons for the expected differences. The first is the time dependent (unsteady) aerodynamic effects related to the motion of the model and the developing jet exhaust plume. The other difference is due to the different model attitudes (relative to the ground plane) required to set a particular angle of attack. The vertical component of velocity inherent in the moving model technique reduces the incidence angle of the model, θ, (in comparison to the static test technique) necessary to achieve a given angle of attack. This reduced incidence angle changes both the impingement angle and the impingement point of the jet on the ground plane resulting in distinctly different plumes in the two test techniques.

The F-15 S/MTD (illustrated in figure 3) is a configuration that is designed to use vectored thrust near the ground (reference 7) and is, therefore potentially sensitive to rate-of-descent modeling when measuring ground effects. Because of this and because it appears that future fighter/attack aircraft will incorporate some form of vectored thrust for control and performance enhancement, the flight test of the F-15 S/MTD was seen as a good opportunity to validate this test technique with a comparison to an actual flight article.

In the present investigation, the ground effect on lift, pitching moment, and drag were measured on a model of the S/MTD (figure 4) while it was subjected to a rate of descent. The model was set in the approach configuration and the aerodynamic loads were measured for several thrust vector angles. These results were then compared to the S/MTD ground effects as determined from the simulation database. This database is composed of data from flight tests of the F-15 aircraft and results of the steady state wind tunnel tests of the S/MTD configuration in and out of ground effects. The procedures used in the wind tunnel tests of the F-15 S/MTD are described in reference 8.

Symbols

All measured aerodynamics were reduced to coefficient form by the methods noted here. All moment data are referenced to the point located at (0.2526) $\bar{c}$ on the model, as noted in figure 5. All work was performed in English units.

b	wing span, in
C_D =	Drag/qS
C_L =	Lift/qS
C_m =	(Pitching moment)/qS$\bar{c}$
ΔC_D	$(C_{D_{\text{Instantaneous}}} - C_{D_\infty})$
ΔC_L	$(C_{L_{\text{Instantaneous}}} - C_{L_\infty})$
ΔC_m	$(C_{m_{\text{Instantaneous}}} - C_{m_\infty})$
$\bar{c}$	mean aerodynamic chord, in
g	Earth gravitational units
h	height of the model over the ground board (referenced to (0.2526)$\bar{c}$, see figure 5), ft
$\dot{h}$	rate of descent, ft/sec
q	dynamic pressure, pounds/ft^2
S	wing area, ft^2
V	velocity, ft/sec
α	angle of attack, degrees
γ	flight path angle (the incidence of the model path relative to the ground plane, degrees)
δ	deflection angle, degrees
θ	incidence angle of the model body axis referenced to the ground plane, degrees

Subscripts:

a	aileron
f	flap
h	horizontal stabilizer
j	jet
l	lower
u	upper
∞	freestream

Abbreviations:

ft	feet
LaRC	Langley Research Center
MAC	mean aerodynamic chord
psf	pounds per square foot
sec	seconds
S/MTD	STOL/maneuver technology demonstrator
sps	samples per second
STOL	short takeoff and landing
VRF	Vortex Research Facility

Facility

The Vortex Research Facility (VRF) (illustrated in figure 6) at the NASA-Langley Research Center was used for the present study. In that facility, the model is suspended on a variable-length strut extending from the bottom of the gasoline-engine powered cart. The strut supports the model, sting, and airline assembly as well as the instrumentation. Angle of attack was changed by pitching the entire strut, sting, and model assembly at the point where the strut was attached to the cart. Velocity was controlled by a cruise-control system on the cart. High-pressure air bottles on the cart provided compressed air for the jets.

For the present test, the test region of the VRF was modified to incorporate a 150-foot long ground plane near the center of the test section. The ground board consisted of two parts: a ramp which was inclined upward 4° for a distance of 100 feet, followed by a horizontal section which extended for an additional 50 feet. The height of the model over the fixed horizontal portion of the ground board was set by adjusting the length of the model support strut. As the model moved horizontally over the inclined portion, the distance from the ground board to the model reduced, thereby simulating an approach along a glide slope of 4°. Rate of descent was dependent on the test velocity as given by the equation:

$$\dot{h} = V_\infty \tan 4°.$$

After moving across the ramp, the model passed over the horizontal section to simulate roll-out or constant altitude flight (see figure 7).

In the VRF twenty four channels of data are transmitted from the cart through a modulated laser to a photo receptor and a mass data storage unit. The channels are sampled at a rate of 111 sps for nearly thirty seconds. The data are then converted to engineering units using an HP-1000 A900 computer. For more information on the data acquisition in the VRF see reference 9.

Model

An 0.083 scale low speed rotary-balance model of an F-15A was modified to match the S/MTD configuration as shown in figures 4 and 5. The model was designed with a q-limit of 6 psf, so 70 ft/sec became the upper limit of testing. This model also had adjustable tail surfaces, flaps, ailerons, and canards. The ranges of motion of these surfaces was sufficient to model properly the approach configuration of the S/MTD aircraft. The configurations tested are detailed in Table I. Landing gear were not modeled in this investigation.

The model was mounted on a blade that entered the top of the model just behind the model reference point (0.2526)$\bar{c}$. The thrust reverser simulator (shown in figure 8) was held in place in the cavity behind the model by two steel bars that attached to each side of the support blade approximately 15 inches above the model. The gaps between the thrust reverser simulator and the model and between the blade mount and the model were bridged by dental dam to prevent flow inside the model.

To insure that the flow on most of the model would be turbulent (similar to the full-scale condition), 1/8 in. wide transition strips of No. 60 Carborundum grit were placed on the model one half inch back from the leading edges of the aerodynamic surfaces and one inch back from the nose.

Reverse thrust simulation was supplied nonmetrically using the thrust reverser simulator described in reference 10 and sketched in figure 8. This device was attached to the sting and positioned (in the cut-out in the model shown in figure 5) such that the position of the reverser nozzles corresponded to the position of the rotating vane thrust reverser on the S/MTD aircraft. The jets were directed by honeycomb inserts embedded in the plenum box cover plate. Different plates directed the jets at different angles. The top reverser ports were blocked during this investigation because of a limitation in the air supply capacity.

Test Procedure

The model was tested at a forward speed of 60 ft/sec in the approach configuration as defined in Table I. The corresponding rate of descent for that test velocity is 4.19 ft/sec.

The vehicle and model were started from rest at one end of the facility and accelerated to the test velocity within 900 feet. The vehicle then passed over the enclosed test section while the model suspended below the vehicle passed through the enclosed test section isolated from the downwash of the vehicle (see figure 6). The test section is approximately 500 feet in length. As the model entered the test section, the model air system was automatically activated and the thrust reverser simulator was powered. This allowed the jets to stabilize before the model passed over the ground board. As the model exited the test section, brakes were applied automatically stopping the vehicle in less than 100 feet.

Results

Data are presented for the approach configuration at a constant q-ratio (q_j/q_∞) of 45. Of principal interest in the data are the results obtained over the inclined portion of the ground board corresponding to the approach phase of landing (i.e., h/b > 0.19).

For this investigation, the top reverser ports were blocked because of a limitation in the air supply capacity. As a result, the aerodynamic coefficients measured are not expected to be the same as those experienced by the S/MTD aircraft during landing. However, it is expected that the increments in the aerodynamic coefficients due to ground effect are valid and useful numbers since the ground effects associated with the top reverser jets should be small in comparison to those associated with the lower jets. For this reason the data are presented as increments from the out-of-ground-effect values of lift, pitching moment, and drag coefficients as functions of ground height.

For the approach condition, the S/MTD configuration is trimmed at $\alpha = 12°$, $\delta_h = 1.88°$, $\delta_c = -12.85°$, and $\delta_a = \delta_f = 20°$ with the upper and lower thrust reversers deflected symmetrically up to a deflection angle of 75°. In addition, the upper reverser will be deflected an additional 10° if additional thrust control is required, however, the lower reverser will not be deflected beyond 75°. The nominal approach thrust vector setting is expected to be 64.3°. This condition, trimmed at $\alpha = 12°$, is noted on a representative lift curve for the F-15 S/MTD in figure 9 and the approach lift coefficient is seen to be approximately 1.1. This information has been taken from the S/MTD simulation database. The configurations tested in this study are defined in Table I.

Moving Model Results - In this investigation, lower reverser angles of $\delta_{j_1} = 45°$, $\delta_{j_1} = 60°$, $\delta_{j_1} = 80°$, and $\delta_{j_1} = 110°$ were tested at a rate of descent of 4.19 ft/sec. The effect of the lower thrust reverser vector angle on the lift coefficient is seen in figure 10. For all jet angles tested, lift coefficient increased slightly above the out of ground effect level beginning at about h/b = 1.0 and remained relatively constant as the model descended to a height of about h/b = 0.5 where the lift began to increase sharply. The reason for this two stage increase is unclear.

For jet vector angles of 45° and 60° (measured from the body axis aft horizontal) the ground effect on lift coefficient increment was essentially the same; lift coefficient increased by about 0.24 by the minimum ground height. For $\delta_{j_1} = 80°$ the increment in lift coefficient was consistently higher at ground heights below h/b = 1.0 than was seen for $\delta_{j_1} = 45°$ and $\delta_{j_1} = 60°$. This is true to an even greater extent at a reverser angle of 110°. Both of these higher thrust reverser angles are beyond the range of thrust vector angles intended to be tested by the S/MTD aircraft on approach. At 110°, unlike the results at the other thrust vector angles, the maximum lift coefficient increase occurred before the model was at the wheel touchdown height, h/b = 0.19. At approximately h/b = 0.22, the lift coefficient increment reached a maximum value of 0.23 then dropped to 0.21 by the touchdown height. Once at this minimum height, the lift continued to drop (as it transitioned to a steady state situation) to a level that was actually slightly below the lift level out of ground effect. Often, aircraft are flown such that sink rate is arrested near the ground. This is referred to as "flaring" the aircraft. These results indicate that such a maneuver would actually allow a lift loss to develop on this configuration and, instead of reducing rate of descent, could actually cause it to increase. As a consequence, the 110° thrust reverser position could only be used if an approach procedure were adopted in which rate of descent was not arrested near the ground.

The effect of lower thrust reverser vector angle on the pitching moment is presented in figure 11. For all thrust vector angles, as the model moved closer to the ground plane and the horizontal stabilizer moved into ground effect, the lift on that surface increased as indicated by the nose-down pitching moment increment. At high thrust vector angles ($\delta_{j_1} = 80°$ and $\delta_{j_1} = 110°$) a ground vortex is able to develop under the horizontal stabilizer and the low pressure associated with a ground vortex induces an incremental nose-up pitching moment. An explanation of the ground vortex can be found in reference 11.

For the three vector angles that span the angles expected to be flight tested (45°, 60°, and 80°), pitching moment coefficient continued to decrease down to the minimum ground height. The magnitude of this negative pitching moment increment increased slightly as the jets were vectored farther forward from $\delta_{j_1} = 45°$ to $\delta_{j_1} = 80°$. When the jets were vectored past 90° to $\delta_{j_1} = 110°$ the ground effect was somewhat different. From h/b = 0.5 down to h/b = 0.28, consistent with the other vector angles, pitching moment decreased, however the magnitude of this decrease was greater than for the other vector angles. At h/b = 0.28, though, the pitching moment began to increase. The pitching moment returned to its freestream level by h/b = 0.25, continued to increase to $\Delta C_m = 0.06$ by touchdown height, then increased more to $\Delta C_m = 0.08$ over the horizontal portion of the ground board.

The effect of lower thrust reverser vector angle on the ground effect on the drag coefficient can be seen in figure 12. As was the case with lift coefficient, drag coefficient remained relatively constant for all vector angles down to about h/b = 1.0. At that height the drag coefficient increased slightly and remained relatively constant as the model descended to a height of about h/b = 0.5 where the drag began to increase rapidly.

For the three vector angles that span the expected flight approach thrust vector range, there was a steady increase in drag coefficient to a ΔC_D level of 0.04 at the minimum height. When the jets were vectored to $\delta_{j1} = 110°$, the drag increase in ground effect was larger than for the aft vectored cases for all heights between h/b = 1.0 and h/b = 0.3. Below h/b = 0.3 the drag increment steadily decreased down to the minimum height where the drag coefficient returned to nearly the freestream level. Once at touch-down height (h/b = 0.19), the drag coefficient continued to decrease to a level of $\Delta C_D = -0.55$ below the freestream level indicating, again, that this vector angle would only be useful if an approach procedure were adopted in which rate of descent was not arrested near the ground.

Comparison to the Simulation Database - There are two major differences between the static wind-tunnel results (procedure described in reference 8) used in constructing the simulation database and the measurements made using the moving model at the VRF. The first principal difference is that the VRF data were obtained while simulating a rate of descent. The other significant difference is that the wind tunnel measurements were made in the presence of a ground boundary layer which has been shown to have a significant impact on the development of the ground vortex created by vectored jets near a ground plane. This impact is detailed in reference 11. In short, the presence of a ground boundary layer allows the ground vortex to penetrate significantly farther upstream (approximately 30 percent) than would be possible in its absence. These two major differences are believed to be the source of the differences discussed below between the two data sets. Other differences between the tests considered less significant are outlined in Table II.

It should also be pointed out that the plots presented for the wind tunnel data at $\delta_j = 110°$ are based largely on interpolations and, to some extent, extrapolations. Reversed thrust is not intended to be used on approach and was therefore, not tested through full height transition. The 110° thrust vector angle was tested out of ground effects and at h/b = 0.35 in the high-lift configuration, however, at wheel-touchdown height it was tested only in the roll-out configuration (i.e., low angles of attack, low lift configuration). The plots represent the best approximation of the simulation database based on that information.

To a lesser extent, the other plots are also based on interpolations of the data used in the simulation database. For example, the wind tunnel data were obtained at $\delta_{j1} = 45°$, 65°, 80°, and 110° whereas the moving-model data were obtained at $\delta_{j1} = 45°$, 60°, 80°, and 110°. For comparison, the wind-tunnel results at $\delta_{j1} = 45°$ and $\delta_{j1} = 65°$ were interpolated to $\delta_{j1} = 60°$.

In figure 13, the lift increment in ground effect for the approach configuration has been plotted for $\delta_{j1} = 45°$ and $\delta_{j1} = 60°$. As height decreases to touchdown height the static wind-tunnel data consistently predicts a greater lift increment due to ground effect than that predicted by the VRF data set. This difference is attributed to the effects of rate-of-descent modeling in the VRF. Once at the minimum ground height for some time, the results from the VRF testing are seen to have the same steady state lift increment levels as those in the wind-tunnel database.

As the thrust reversers were vectored farther forward, the presence of the ground boundary layer is seen to have a greater effect. This is illustrated in figure 14. For both $\delta_{j1} = 80°$ and $\delta_{j1} = 110°$ not only is the lift increment different as h/b reduces to the minimum ground height, but also, the steady state levels measured once the models were at that minimum height are different. The reason for the differences at the minimum ground height is believed to be due to the presence of a ground boundary layer in the wind tunnel testing. The differences at the other ground heights are due to both rate-of-descent modeling in the VRF and the presence of a ground boundary layer in the wind tunnel testing - these two effects can not be separated for this particular set of data.

The differences in pitching moment are similarly illustrated in figures 15 and 16. The thrust reverser configurations $\delta_{j1} = 45°$ and $\delta_{j1} = 60°$ are shown in figure 15. At $\delta_{j1} = 45°$, much like the results seen for the lift coefficient, the wind tunnel results predicts greater nose-down pitching moment than the dynamic measurements from the VRF as the model height is reduced to the minimum ground height. However, once at that height for some time, and the VRF flowfield transitions to a steady state, the level of nose-down pitching moment measured by the two techniques are nearly equal. Again this difference at heights greater than that corresponding to wheel touchdown is attributed to the modeling of a rate of descent in the VRF testing.

As the thrust reverser jet is vectored further to $\delta_{j1} = 60°$, the comparison is similar down to a model height of approximately h/b = 0.3. Below that height the wind-tunnel database indicates that the configuration experienced progressively less nose-down pitching moment as the model approached the ground. This is, again, believed to be due to the presence of the ground boundary layer in the wind tunnel testing. This boundary layer allows the thrust reverser jets to penetrate farther upstream before forming the ground vortex. In this situation it is believed that the ground vortex has developed under the horizontal stabilizer and the low pressure vortex has reduced the lift on that surface. Greater penetration of the ground vortex should also induce greater upwash at the wing. The net effect would be as is seen in figures 13 and 15: increased steady state pitching moment increment and no difference in lift increment between the VRF data and the wind-tunnel database.

The effect of the ground boundary layer is even more pronounced as the thrust reverser jets are vectored further forward. This is presented in figure 16. In these configurations, more upwash is indicated at the canard in the wind tunnel database than was indicated in the VRF results because the ground vortex could not penetrate as far upstream in the absence of a ground boundary layer.

Similar results were found in the drag measurements as shown in figures 17 and 18. Again, at δ_{j1} = 45°, where the jets are blown well aft, the presence of the ground boundary layer in the wind tunnel test had little effect on the steady state aerodynamics, but, as the thrust reverser was directed progressively farther forward, the boundary-layer effect was intensified as was seen in both lift and pitching moment. For all settings, a significant effect is evident due to rate-of-descent modeling in the VRF at all model heights greater than the minimum height.

Conclusions

A moving-model ground-effects testing method has been used to study the influence of rate-of-descent on the approach aerodynamics of the F-15 S/MTD configuration. A comparison of the results obtained while modeling rate of descent to the predictions of the simulation database (predictions based on data obtained in wind tunnel with zero rate of descent) showed significant differences due both to the rate of descent in the moving-model test and to the presence of the ground boundary layer in the wind-tunnel test. Relative to the simulation database predictions, the rate-of-descent modeling produced substantially less lift increase in ground effect, less nose-down pitching moment, and less increase in drag. These differences became more prominent at the larger thrust vector angles.

The results of this investigation indicate no safety-of-flight problems with the lower reverser vectored up to 80° on approach. They also indicate that this configuration could employ a nozzle concept using lower reverser vector angles up to 110° on approach if an approach procedure were adopted in which rate of descent was not arrested near the ground and if inlet reingestion were found not to pose a problem.

Acknowledgments

The author would like to thank Garrett Billman of McDonnell Douglas Corporation for providing the predictions on the aerodynamic characteristics of the F-15 S/MTD used for comparison in this report. These predictions were made based on the simulation database.

References

1. Innis, R. C.; Anderson, S. B.: Comparisons of Simulator and Flight Results on Augmentor-Wing Jet STOL Research Aircraft. Presented at the NASA Conference on STOL Technology, NASA SP-320, pp. 283-290, 1960,.

2. Baker, P.; Schweikhard, W.; Young, W.: Flight Evaluation of Ground Effect on Several Low-Aspect-Ratio Airplanes. NASA TN-D-6053, October 1970.

3. Campbell, J. P.; Hassel, J. L.; Thomas, J. L.: Ground Effects on Lift for Turbofan Powered-Lift STOL Aircraft. Journal of Aircraft, Vol. 15, No. 2, pp. 78-84, February 1978.

4. Pelagatti, C.; Pilon, J.; Bardaud, J.: Analyse Critique Des Comparisons Des Resultats De Vol Aux Previsions De Soufflerie Por Des Avions De Transport Subsonique Et Supersonique. AGARD-CP-187, Flight/Ground Testing Facilities Correlation, 1975.

5. Chang, R.: An Experimental Investigation of Dynamic Ground Effect. Dissertation, University of Kansas, Lawrence, 1985.

6. Kemmerly, G. T.; Paulson, J. P., Jr.: Investigation of a Moving-Model Technique for Measuring Ground Effects. NASA TM-4080, January 1989.

7. Roberts, F. D.: The F-15 STOL and Maneuver Technology Demonstrator (S/MTD) Program. SAE 87-2383, December 1987.

8. Blake, W. B.: F-15 SMTD Low Speed Jet Effects Wind Tunnel Test Results. NASA CP-10008, pp. 91-119, 1988.

9. Satran, D.; Neuhart, D.; Holbrook, G.: Vortex Research Facility Improvements and Preliminary Density Stratification Effects on Vortex Wakes. AIAA 85-0050, January 1985.

10. Joshi, P.: Generic Thrust Reverser for Near Term Application. AFWAL TR-84-3094, February 1985.

11. Stewart, V.; Kemmerly, G. T.: Characteristics of the Ground Vortex Formed by a Jet Moving Over a Fixed Ground Plane. AIAA 89-0650, January 1989.

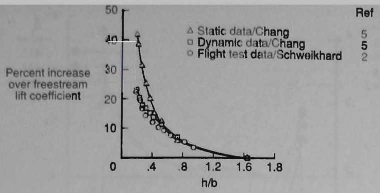


Figure 1. Static, dynamic, and flight test data from an XB-70 at α = 9.3°.

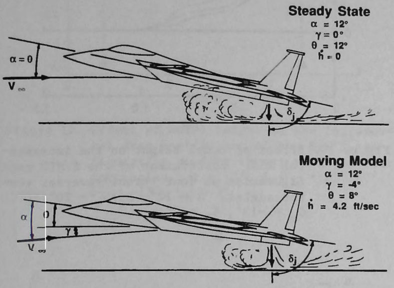


Figure 2. Schematic of dynamic and steady state ground effects.

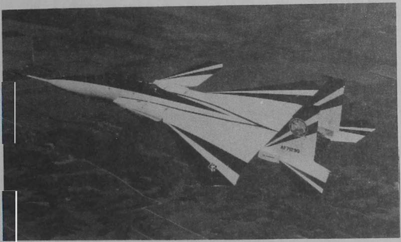


Figure 3. F-15 S/MTD aircraft.

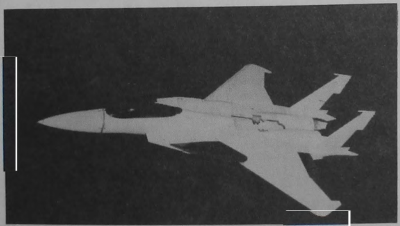


Figure 4. S/MTD model tested in the VRF.

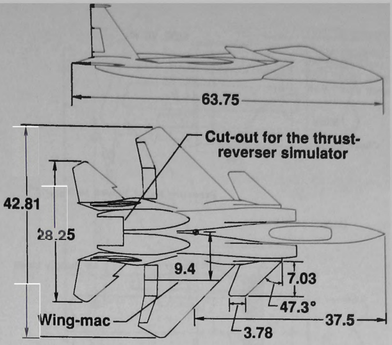


Figure 5. Sketch of the F-15 S/MTD model. Dimensions are in inches.

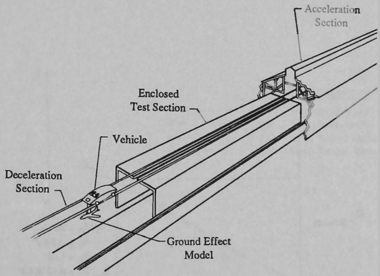


Figure 6. Schematic diagram of the VRF.

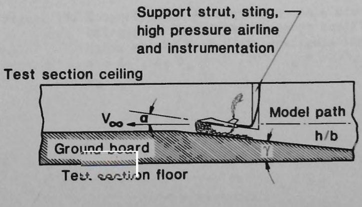


Figure 7. Model passing through the test section in the VRF.

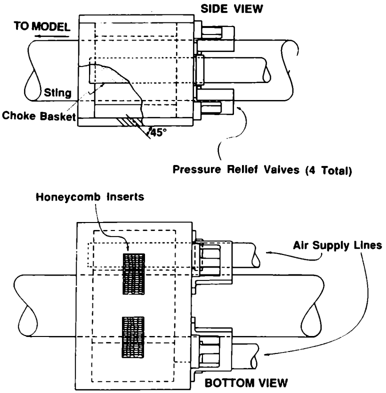


Figure 8. Sketch of the thrust reverser simulator.

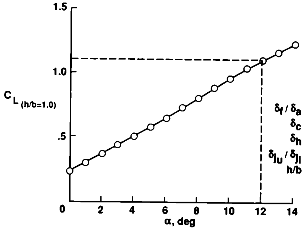


Figure 9. Lift curve of the F-15 S/MTD in approach configuration.

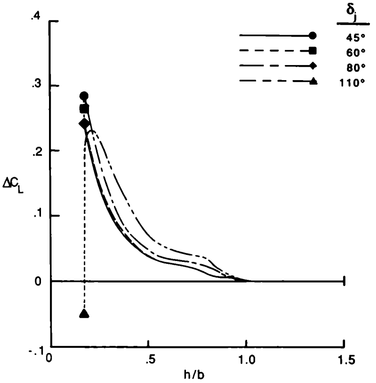


Figure 10. Effect of model height on the incremental lift coefficient of the S/MTD configuration at four thrust reverser vector angles. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

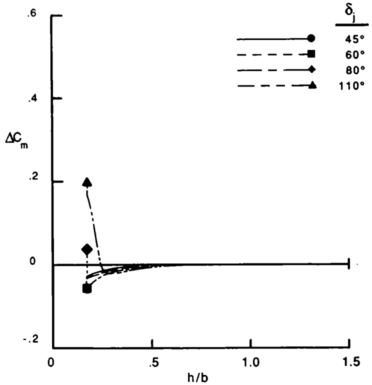


Figure 11. Effect of model height on the incremental pitching moment coefficient of the S/MTD configuration at four thrust reverser vector angles. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

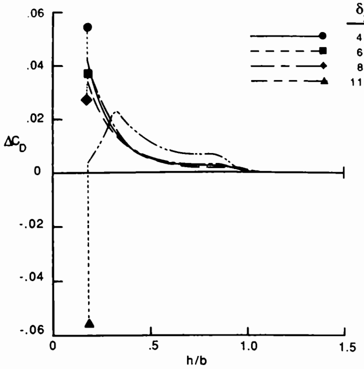


Figure 12. Effect of model height on the incremental drag coefficient of the S/MTD configuration at four thrust reverser vector angles. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

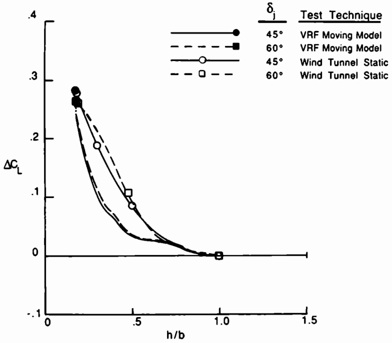


Figure 13. Comparison of static and dynamic ground effects on the lift coefficient of the F-15 S/MTD configuration. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

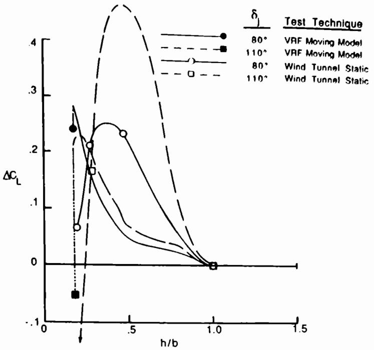


Figure 14. Comparison of static and dynamic ground effects on the lift coefficient of the F-15 S/MTD configuration. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

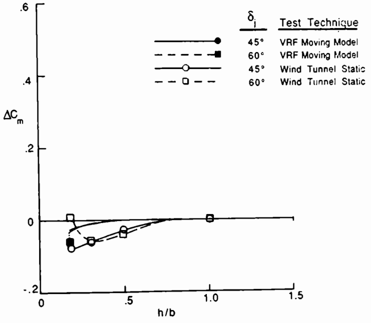


Figure 15. Comparison of static and dynamic ground effects on the pitching moment coefficient of the F-15 S/MTD configuration. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

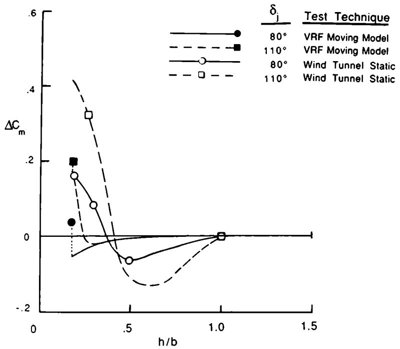


Figure 16. Comparison of static and dynamic ground effects on the pitching moment coefficient of the F-15 S/MTD configuration. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

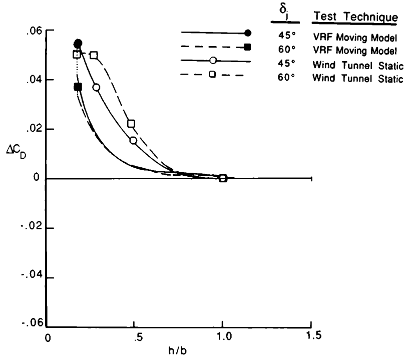


Figure 17. Comparison of static and dynamic ground effects on the drag coefficient of the F-15 S/MTD configuration. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

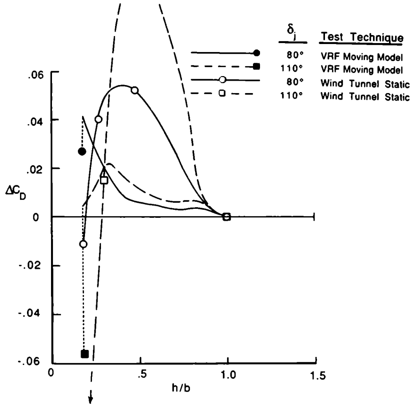


Figure 18. Comparison of static and dynamic ground effects on the drag coefficient of the F-15 S/MTD configuration. $\alpha = 12°$, $\delta_f/\delta_a = 20°/20°$, $\delta_c = -13°$, $\delta_h = 2°$

89-3282-CP

DEVELOPMENT OF AN AUTONOMOUS LANDING GUIDANCE SYSTEM SIMULATION MODEL

James M. Zeh   Daniel L. Young   Norman J. Couture Jr.

Wright Research and Development Center
Flight Control Division (FIG)
Wright-Patterson AFB, Ohio

Abstract

This paper addresses the development and application of an Autonomous Landing Guidance (ALG) simulation. The autonomous landing scenario requires a pilot to locate, designate, and land on a bomb damaged runway using only on-board systems. The need for simulating the ALG system arose from the United States Air Force, Short Takeoff and Landing (STOL)/Maneuver Technology Demonstrator (S/MTD) Program. To accomplish this task the high fidelity S/MTD aircraft simulation was used in the Large Amplitude Multi-mode Aerospace Research Simulator (LAMARS). Approximately 120 hours of pilot-in-the-loop evaluations were conducted to validate and optimize the ALG simulation. This evaluation was part of a complete analysis of the operational flight control laws conducted at the Flight Dynamics Laboratory (FDL). Recommendations resulting from this testing were implemented in the S/MTD aircraft flight control computer before actual ALG flight tests.

Background

Program Objectives/Origin

In any major conventional conflict hostile forces will attempt to destroy friendly airfields to cripple air operations and gain air superiority. Therefore, friendly aircraft must expect to encounter bomb damaged airfields upon return to home base. To counter this threat a Statement of Operational Need was released for the development of a high performance aircraft which could land and takeoff from a bomb damaged runway. The S/MTD Program evolved to meet these needs.

The primary objective of the S/MTD Program is to develop and flight demonstrate four technologies on a modified F-15B aircraft. The four technologies are 2-Dimensional thrust vectoring/ reversing nozzles, Integrated Flight/Propulsion Control (IFPC), rough/soft field STOL landing gear, and advanced Pilot Vehicle Interface (PVI).

A contractual requirement of this program is to demonstrate the capability to takeoff and land on a bomb damaged runway using only on-board sensors (no active ground guidance aids). Additionally this must be accomplished in adverse weather (.5 mile visibility and 200 foot ceiling) and on a wet/repaired runway containing a 50 by 1500 foot Minimum Operating Strip (MOS). Another contractual requirement specifies the S/MTD aircraft demonstrate this capability with no degradation to the air-to-air maneuverability and range of the baseline F-15B aircraft.

In support of the S/MTD Advanced Development Program Office (ADPO), the Flight Control Division of the Flight Dynamics Laboratory was tasked to develop a simulation to verify that requirements were satisfactorily met and to address safety of flight issues before first flight of the aircraft. Over a four year period a series of S/MTD simulations were developed at FDL and McDonnell Aircraft (McAir). The simulations were updated as wind tunnel data became available and control laws were developed. Numerous studies were completed to ensure a flying qualities rating of level 1 for the aircraft. The ALG evaluations were conducted upon completion of the flying quality and PVI studies.

Approach

To successfully demonstrate the ALG system, the FDL simulation team had to accomplish several tasks. First the high-fidelity S/MTD simulation was developed as described above. Then the LAMARS cockpit was modified with controls and displays necessary to complete the ALG task. Third, the S/MTD APG-70, Synthetic Aperture Radar (SAR) was researched to determine requirements to emulate the system. Required hardware was purchased and software developed to simulate the SAR. Finally, a modified ALG scenario was developed to maximize APG-70 performance.

The remainder of this paper extensively addresses the development, operation, and evaluation of an ALG simulation system at FDL. Discussed to a lesser degree are the test apparatus and computer architecture in which the ALG system was evaluated.

S/MTD Simulation Architecture

The S/MTD simulation computer network is built around two Gould SEL 32/77s, a SEL 32/27, and a SEL 32/97. These are 32 bit digital machines with high speed, real-time and scientific computing capability. A generic block diagram of the simulation is shown in Figure 1. The aerodynamic, flight control system, actuator models, landing gear model, Head Up Display (HUD) algorithms, and environment software are updated 40 times per second. The head down display algorithms and propulsion model are updated 20 times per second.

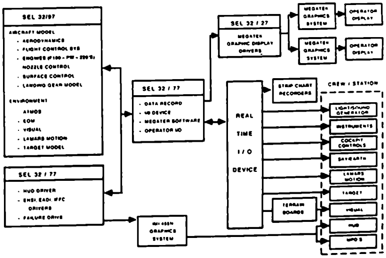


Fig. 1 Generic Block Diagram

Simulator

All experiments were conducted in the Large Amplitude Multi-Mode Aerospace Research Simulator. The LAMARS consists of a single-seat cockpit installed in a 20 foot diameter dome attached to the end of a 30 foot beam. The combined beam/dome movements result in a five degree-of-freedom motion system which can generate angular velocities of up to 60 degrees per second, linear velocities of up to 13 feet per second, and instantaneous accelerations of up to 3 g's.

Visual Display System

The visual scene generated for the landing evaluations is from an American Airlines/Redifon 1500:1 scaled terrain board system. The viewing area is continuous in heading and roll, but limited in pitch from +24 to -47 degrees. This imagery is projected in the LAMARS dome as a fixed forward 60 degree diagonal field of view. The runway area on the terrain board was modified with an overlay portraying bomb damage to establish five 50 x 1500 foot Minimum Operating Strips (MOS's)

Crew Station

The LAMARS cockpit represents to the extent possible the forward crew station of the S/MTD aircraft (Figure 2). An F-16 Wide Field Of View (WFOV) Head Up Display (HUD) was used instead of the F-15E WFOV HUD. Both HUD's have a 20 degree by 30 degree field of view. Two 6 inch Multi-Purpose Displays (MPD's) provided heads down display information. The S/MTD center stick and rudder characteristics were provided by a McFadden three axis feel system. Also provided was the S/MTD throttle quadrant which has a spring loaded thrust reversing region. A photograph from the pilot's eye inside the LAMARS cockpit is shown in Figure 3. For further information on how the S/MTD simulation was accomplished refer to reference 1.

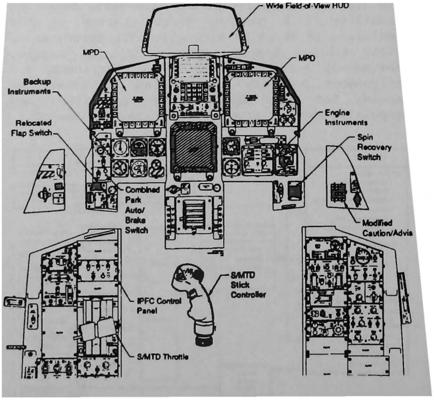


Fig. 2 S/MTD Forward Crew Station

Fig. 3 S/MTD LAMARS Cockpit Photograph

APG-70 Simulation Description

APG-70 SAR Image Development

The development of an APG-70 Synthetic Aperture Radar image was paramount to successfully simulate the ALG task. Since the visual cueing in the LAMARS is provided by an American Airlines/Redifon terrain board system, methods to obtain SAR imagery of the terrain were determined. A system that could process a SAR image real-time would have been optimal; however, the expense for such a system was greater than allotted. The best solution found was to create a library of digitally processed still photos.

To provide a realistic bomb-damaged runway, a magnetic overlay for the existing runway area was developed. Significant effort was expended in obtaining the probable bombing pattern used for runway denial. The bomb craters were positioned to produce several possible MOS's and scaled to represent damage done by conventional munitions.

In order to provide a realistic SAR display, the operation of the APG-70 was researched. To support the ALG task, only the stabilized High Resolution Map (HRM) mode with .67 and 1.3 nautical mile (nm) map scales was required. As illustrated in Figure 4, the maps are stabilized to a fixed point on the ground and rotate around their center based on aircraft position. The fixed point is selected by the pilot as described in the ALG Task Description section of this paper.

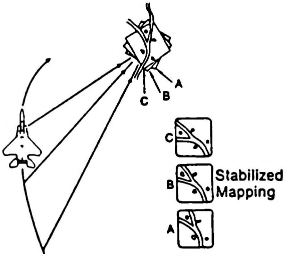


Fig. 4 Stabilized Mapping

To model this mode of APG-70 operation in a cost effective manner a grid consisting of thirty four fixed points centered around the touchdown location was chosen. Then, an algorithm was developed to select the nearest available point on the grid. Because of stabilized mapping, a series of still photos was necessary for each aimpoint. The number of required photos was determined by the desired flight path. Thirty six digitized photos, in 2 degree increments, were taken for each aim point yielding a total of 1224 per map scale.

A block diagram of the equipment used in producing the still photos is shown in Figure 5. The stills were stored via a Panasonic video disc recorder/player. As seen in Figure 5, the still photos were then digitally processed using an IBM AT configured with a set of Data Translations boards. The processing of the video stills was done by Mr. Mark Sturgell of the Avionics Laboratory, Wright Research and Development Center, Wright-Patterson AFB, Ohio. Although the digitizing process is an important step in the development of the APG-70 SAR image, it is not addressed here. If further information is needed, the digitizing process is extensively addressed in reference 2. An example of the processed pseudo-SAR image is shown in Figure 6.

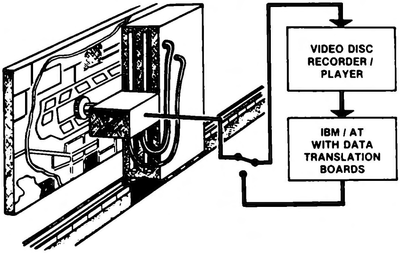

VIDEO DISC RECORDER / PLAYER
IBM / AT WITH DATA TRANSLATION BOARDS

Fig. 5 Equipment Configuration for Producing Still Photos.

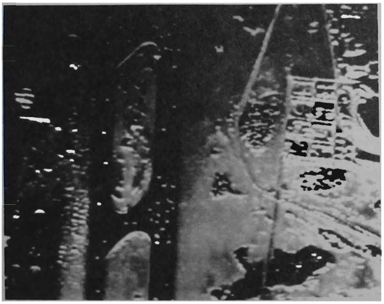


Fig. 6 Processed pseudo-SAR Image

Each processed pseudo-SAR frame was then systematically stored using the video disc recorder. The video disc was chosen as the medium for storing the SAR images primarily for the storage and retrieval capabilities of the recorder/player. The video recorder/player provided the systematic storage and computer controlled real-time retrieval capabilities that were essential for the ALG simulation system.

APG-70 Simulation Architecture

The architecture of the APG-70 simulation is shown in the block diagram of Figure 7, which will be referenced throughout the following discussion. The LAMARS is configured with a mock-up cockpit of the S/MTD aircraft and provides the interface between the APG-70 simulation and the pilot. Pilot inputs from the cockpit are converted to digital values and passed to SEL "A", a Gould Sel 32/77. These digital values are then stored in a buffer located in shared memory.

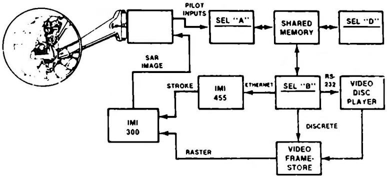


Fig. 7 APG-70 SAR Simulation Architecture.

SEL "D", a Gould SEL 32/97, receives this information through the SEL shared memory interface. S/MTD simulation software calculates aircraft states using these values. This information is then transferred to SEL "B", a Gould SEL 32/77, through the shared memory interface.

SEL "B" contains the software required to control the stroke symbology and raster video on the MPD. First, the SAR frame number is determined by SEL "B" algorithms. Then, the SAR image is updated on the video disc player by transferring the necessary commands and frame number via an RS-232 interface. The video disc player's output is connected to a modified Digital Video Systems Corporation, Timebase Corrector/Framestore. The framestore is a two buffer video device which is used to eliminate blanking of the SAR image during the search mode of the video disc player. SEL "B" informs the framestore to switch between the two buffers at the appropriate time by applying a discrete signal. The stroke symbology displayed on the MPD is generated by the IMI 455 graphics computer. The necessary information to control the stroke symbology is transferred from SEL "B" through an Ethernet interface. Finally, the IMI 300 superimposes stroke symbology on the SAR image and outputs this to the MPD.

APG-70 Display Format

APG-70 SAR display format is dependent on the radar mode of operation. The ALG task requires only the High Resolution Map (HRM) and the Precision Velocity Update (PVU) modes of the APG-70. The HRM mode of the APG-70 uses a mapping or targeting display format. The targeting format is explained later in the ALG Task Description section. The mapping format and the PVU display are described below Figures 8 and 9 respectively.

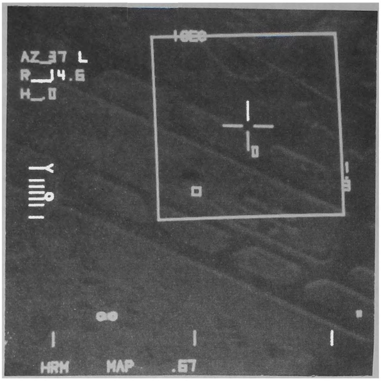


Fig. 8 HRM Mapping Display Format

The figure above is the mapping display format of the HRM mode. This mode is used to locate and designate the Touchdown Point (TDP) and heading of the Minimum Operating Strip (MOS). Symbology description begins at the lower left corner and proceeds in a counter clockwise direction.

The HRM readout in the bottom left corner of the display indicates the APG-70 is in high resolution mapping mode. To the right is the cursor function readout, which currently reads MAP. In this format the cursor, which is described later in this section, is used to locate a new radar aimpoint. The radar aimpoint is a location on the ground that the radar sweep is centered about. It is always located in the middle of the HRM display.

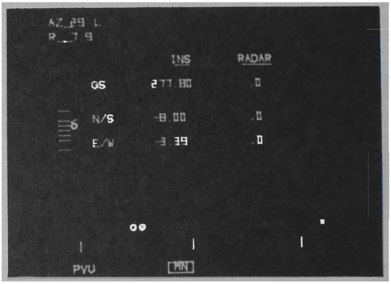
The next two readouts deal with scale of the radar display. The simulated HRM mode uses 1.3 and .67 scaled map sizes. The 1.3 map is available when the aircraft is within 40 nautical miles (nm) of the aimpoint. A .67 map is available within a range of 18 nm. The current map size readout (1.3) is displayed halfway up the right side of Figure 8. This means a square map representing 1.3 nm is displayed on the MPD. The .67 readout at the bottom of Figure 8, indicates display window size. When the next aimpoint is requested, a map .67 by .67 nm is drawn on the MPD.

The seconds remaining to generate a new patch map are displayed at the top of Figure 8. A new map is created each time the readout counts down to zero. Typically this takes 3 seconds.

At the upper lefthand corner of the figure, there are three alpha-numeric readouts. The top readout is an azimuth to aimpoint indicator, it displays the horizontal angle (in degrees) between the aircraft's inertial heading vector and the aircraft's radar to aimpoint vector. Below this is the range to aimpoint readout, it indicates the straight line distance between the aircraft and the aimpoint. The third readout represents the angle of the designated MOS (from North) in degrees and is called the designated heading readout. Designation is discussed later in the ALG Task Description section.

Continuing down on the left of the SAR display, the radar antenna elevation angle is indicated. The caret symbol moves against a fixed scale consisting of tick marks at ± 1, ± 2, and ± 5 degrees. The azimuth limit indicator is located below and slightly to the right of this scale. As illustrated in Figure 8, the current azimuth scan is depicted by a caret traveling between two small circles. These circles indicate the azimuth radar scan for updating the patch maps of the current aimpoint.

The display window is the large box drawn in the upper right portion of Figure 8. This box appears when a different resolution map is requested by the pilot. The window encompasses the area that will be displayed when a new map is constructed. The cursor symbol is located in the center of the display window. In the mapping mode the cursor position provides coordinates for a new radar aimpoint.



	INS	RADAR
GS	277.80	.0
N/S	-8.00	.0
E/W	-3.39	.0

Fig. 9 PVU Mode.

Shown above is the display format of the PVU mode. This mode of the APG-70 SAR is used to update Mission Navigator (MN) velocities or the Inertial Navigation System (INS) velocities. Due to the amount of time needed to perform an actual PVU, the PVU mode implemented in this ALG system is modified to apply a bias to the INS velocities. When the PVU mode is activated the pseudo-SAR display in the MPD is replaced with the display shown in Figure 9. The INS column displays the velocities from the INS and the RADAR column displays the radar velocity errors. The INS and radar error velocities correspond to the Ground Speed (GS), North/South (N/S) speed, and East/West (E/W) speed in feet per second. Initially only the INS velocities are displayed. Radar velocity errors take approximately 8 seconds to calculate and display. If the radar velocity errors are decreasing in value, they are accepted by the pilot, and three seconds later the APG-70 returns to the HRM mode. If they are not accepted, the APG-70 will automatically return to HRM mode within 12 seconds.

HUD Format Description

The gear down HUD format is shown in Figure 10. This format is used when the aircraft is on final approach. The navigation steering mode selected is Autonomous Landing (AUTL). Unique HUD symbology is described below. Other HUD formats are briefly discussed in the ALG Task Description section of this paper.

The MOS symbology is a 50 by 1500 foot representation of the landing strip. The MOS is drawn with respect to the designated touchdown point and heading.

The E-bracket is used by the pilot in conjunction with the throttle to acquire the required landing airspeed. The E-bracket is driven with respect to the referenced Angle Of Attack (AOA). When the aircraft is at the referenced AOA the E-bracket center line is aligned with the velocity vector's wing tips. On the other hand, if the current AOA is above the referenced AOA the E-bracket would be below the velocity vector prompting the pilot to throttle forward to increase power.

The carets are fly to commands driven with respect to the velocity vector and azimuth/elevation errors. The azimuth error is based on the horizontal distance from the aircraft to the desired ground track, and the elevation error is based on the aircraft's vertical distance from the desired glideslope. The flight director algorithm drives the carets in elevation to provide a 3 degree glideslope originating from a point 60 feet back from the designated touchdown point. The carets are driven in azimuth to steer the aircraft to the designated heading. The carets rotate indicating optimal bank angle for obtaining the designated MOS heading.

The flare cue provides information to flare the aircraft on a 2 degree glideslope. The flare cue appears when the radar altitude is less than 200 feet. Initially, the flare cue should be below the velocity vector indicating glideslope is greater than 2 degrees. When the aircraft obtains a 2 degree glideslope, the flare cue aligns with the velocity vector.

The IFPC mode readout displays the current flight control system mode selected. The system has several modes of operation to enhance flying qualities and reduce pilot workload. The different modes are in-flight selectable and each will be discussed as needed. The IFPC system is currently in SLAND mode, as shown in Figure 10. This mode is a flaps down mode which utilizes the unique characteristics of the S/MTD aircraft. In SLAND mode, aircraft speed is held constant for a given throttle setting, while airspeed and pitch attitude are decoupled. A detailed description of how the S/MTD aircraft operates is beyond the scope of this paper. Further information can be obtained from reference 1.

Two additional readouts also appear in the HUD under certain conditions. The autobrake advisory readout will flash for 5 seconds if the autobrake system disengages. The other readout is the thrust reversing vane advisory readout which illuminates when the thrust reversing vanes are deployed.

EHSI Display Description

The Electronic Horizontal Situation Indicator (EHSI) is displayed on the right MPD and is shown in Figure 11. The aircraft symbol is fixed at the center of the display with the nose of the aircraft always pointed to the top of the display. The compass card symbology rotates about the aircraft symbol to the current heading of the aircraft as indicated by the lubber line.

There are two different symbols used to represent the waypoints. The small circle represents the relative position of the current steer point with respect to the aircraft. The other waypoints are indicated by numbered triangles and they are positioned to indicate the desired flight path.

The range scale readout indicates the distance in nautical miles from the center of the aircraft to the perimeter of the compass card. The possible range scale values are 10, 20, 40, 80, and 160 nm. On either side of the range scale readout there is a triangle associated with the adjacent bezel switch. The range scale readout is changed by depressing the bezel switch above the appropriate triangle.

The three alpha-numeric lines located in the bottom right portion of the display are discussed next. These provide additional steering information to the pilot. The steer point readout indicates which waypoint the flight director is steering towards. The next line down indicates the commanded heading in degrees and the range in nautical miles to the current steer point. The third alpha-numeric line represents the intersect time. This is the time (hours:minutes:seconds) remaining until the aircraft reaches the current steer point.

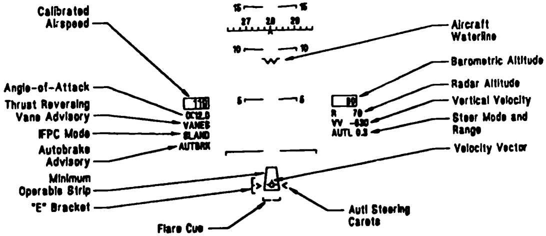


Fig. 10 HUD Display with AUTL Selected and Gear Down

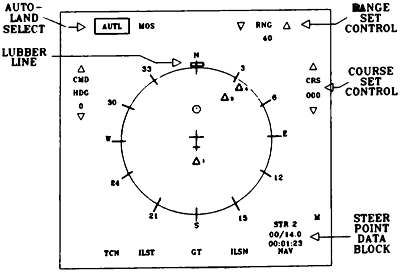


Fig. 11 EHSI Display with Undesignated TDP.

ALG Task Description

In the following discussion the autonomous landing scenario shown in Figure 12 is used to describe the ALG task. The scenario begins approximately 26 nautical miles from the airfield. The waypoints were selected to provide optimal look angles for the APG-70 radar mapping. The S/MTD simulation is initialized at an altitude of 5000 feet, an airspeed of 350 knots, and positioned at waypoint 1. The radar system is in the HRM mode with a 1.3 map centered on the control tower. This map is on the left MPD and updated every three seconds. The pilot must locate, designate, and land on a bomb damaged runway using only on-board systems, while maintaining a preprogrammed flight path. The following discussion breaks the ALG task into 5 parts.

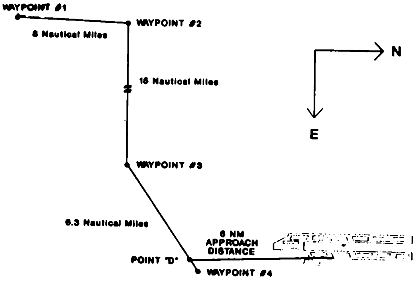


Fig. 12 ALG Scenario

Waypoint 1 To Waypoint 2

During this segment of the ALG scenario the pilot must perform several tasks. First, he must use the APG-70 to obtain a new aimpoint. Then he needs to change the range scale of the EHSI to 40 nautical miles. Additionally, the pilot must periodically view the HUD to maintain proper flight path and descend to 3,500 feet. Each task is described in detail below.

The pilots initial task is to acquire a new aimpoint using the SAR. First, he selects the left MPD as the display of interest which allows the center stick and throttle switches to control the APG-70. Using these switches in conjunction with the MPD, the pilot generates a new map centered about the desired landing area.

Another task accomplished during the flight from waypoint 1 to waypoint 2 is increasing the range scale on the EHSI. The EHSI is initialized in AUTL steering mode with a range scale of 10 nautical miles. The pilot increases the range scale to 40 nautical miles to enable all waypoints to be displayed. This is accomplished as described in the EHSI Display Description section.

The pilot periodically views the HUD to maintain flight path while obtaining the desired altitude. The HUD is shown in Figure 13 and unique symbology is described below.

The azimuth command steering bar is a fly to command. By keeping it aligned with the velocity vector the pilot maintains a bearing towards waypoint 2. When the aircraft is within a 2 nautical mile radius of waypoint 2 and the range to it starts to increase, the flight director steering algorithm will sequence to waypoint 3. The flight director azimuth steering bar commands a maximum 45 degree bank towards waypoint 3.

The IFPC mode readout is currently in Cruise mode. This flaps up enhanced mode utilizes the thrust vectoring/reversing capabilities of the demonstrator to provide optimal flight path control. For further information on the S/MTD IFPC system refer to reference 1.

The ALG block data is located on the right bottom portion of the HUD format shown in Figure 13. The steering mode readout (NAV) indicates the flight director steering commands currently displayed. To the right of this is the current steer point number. This is the waypoint number that the azimuth steering bar is guiding the aircraft towards. The range to steer point is displayed directly below the steering mode readout and represents the range in nautical miles from the aircraft to the steer point position. The next line of block data indicates the time (hours: minutes:seconds) remaining for the aircraft to intersect the steer point.

Waypoint 2 To Waypoint 3

The pilot must accomplish several tasks during this segment. After obtaining the proper heading to waypoint 3 the pilot periodically views the HUD to maintain desired flight path. Then he uses the APG-70 to complete a Precision Velocity Update (PVU), acquire a .67 map, and designate the touchdown point (TDP). Before reaching waypoint 3 the pilot must obtain an altitude of 1800 feet and an airspeed of 250 knots. Each task is described in detail below.

After the pilot establishes a heading towards waypoint 3, he puts the APG-70 in the PVU mode. The PVU is accomplished as described in the APG-70 Display Format section of this paper. Upon returning to HRM mode the pseudo-SAR frame symbology is adjusted to correspond with the radar velocity errors accepted during the PVU.

Now the pilot directs most of his attention to the task of locating and designating a useable section of the bomb damaged runway. He uses the

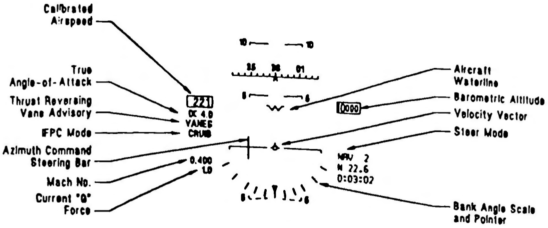


Fig. 13 HUD with Gear Up and NAV Steering

SAR display in conjunction with switches on the control stick and throttle to accomplish this. First, he requests a new pseudo-SAR patch map centered on a possible landing area. If the aircraft to aimpoint range is less than 18 nautical miles, a .67 map would be requested. Approximately 3 seconds later, a new radar aimpoint is established and a new patch map displayed. Regardless of which resolution map is used, the designation process and SAR symbology are the same.

The pilot selects TARGET mode after the new SAR frame is displayed. This mode shown in Figure 14 has two important differences from MAP mode. The FREEZE readout indicates the APG-70 is no longer updating SAR images. However, the azimuth, range, and heading readouts continue to update. The second difference is the target cursor. This symbol has a triangle, centered on the original map cursor.

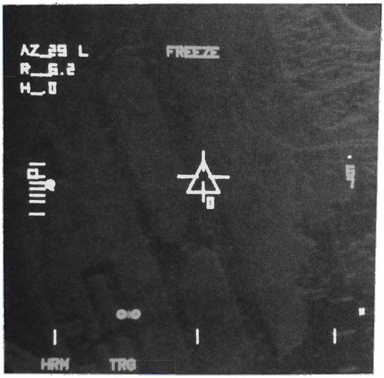


Fig. 14 APG-70 Target Mode

The pilot designates a suitable TDP on the displayed bomb damaged runway using the target cursor. This produces the SAR symbology format shown in Figure 15. The DESIGNATE readout and the boxed TRG function are illuminated for five seconds to indicate the TDP has been designated. Another change in the SAR format is the pattern-line steering symbol. It is initialized at a zero degree heading as indicated by the readout directly below the target cursor. The pattern-line steering symbol is scaled to represent the required 1500 foot landing distance when displayed on the .67 map.

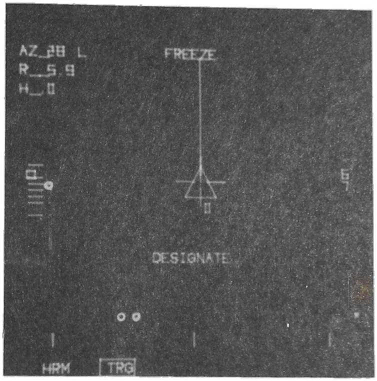


Fig. 15 TARGET Mode with Pattern-Line Steering

The next step in the designation process is for the pilot to designate the desired MOS heading. First he enables the pattern-line steering symbol as shown in Figure 16. The arrowhead indicates the symbol can be rotated about the center of the target cursor to select a MOS heading. The heading readout below the target cursor will update as rotation occurs. The pilot's objective is to align the pattern-line steering symbol with a useable section of the bomb damaged runway on the SAR image.

Once the pilot is satisfied with the position of the pattern-line steering symbol he designates the MOS heading. Figure 17 shows the TARGET mode symbology format after designation. There are two main differences to the display. First, the arrow on the pattern-line steering symbol is replaced with a circle. Secondly, the DESIGNATE readout and boxed TRG function are illuminated for five seconds. This indicates the pilot has completed the designation process.

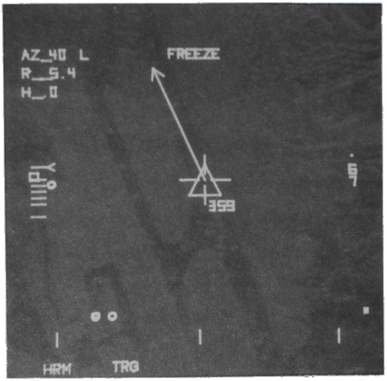


Fig. 16 TARGET Mode with Pattern-Line Steering Enabled.

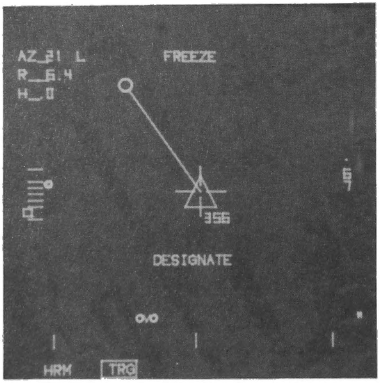


Fig. 17 TARGET Mode with Designation Completed.

After heading is designated, the EHSI display provides additional situation awareness to the pilot. Figure 18 shows the EHSI format after designation. The new format incorporates the MOS ground track symbol which is made up of a box and an arrow. The box is centered on the designated TDP and the arrow overlays the MOS final approach ground track. The length of the arrow represents 8 nm on the EHSI display.

The pilot's final APG-70 task in this phase is to select HRM mode on the SAR display. This allows the APG-70 to return to the stabilized mapping mode. When the aircraft is within a 2 nautical mile radius of waypoint 3 and the range to it starts to increase, the flight director steering algorithm will sequence to waypoint 4. The flight director azimuth steering bar commands a maximum 45 degree bank towards waypoint 4.

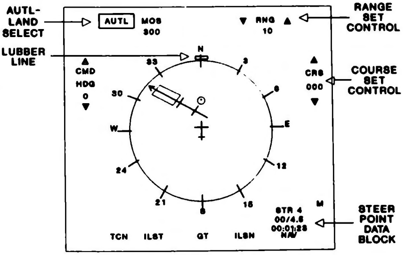


Fig. 18 EHSI with Designation Complete.

Waypoint 3 To Ground Track

The pilot workload is fairly heavy during this phase of the scenario. This is due to the pilot having approximately 100 seconds to complete the following tasks. First, the pilot obtains a flight path towards waypoint 4. Once the flight path is established, he performs a final PVU. Based on these results the MOS heading and TDP may need to be redesignated. Next, the pilot selects the HUD as display of interest. This changes steering from NAV to Bankline providing steering cues towards the touchdown point. During this phase the pilot must obtain an airspeed between 230-235 knots while maintaining an altitude of 1800 feet. Each task is described in detail below.

Once the pilot has established a bearing towards waypoint 4, a final PVU is performed at a range of 4 to 5 nautical miles from the final approach ground track. Upon return to the HRM mode the pilot selects TARGET function on the APG-70. If the PVU errors are significant, the pattern-line steering and target cursor will not align with the previous designation. Consequently, the pilot may need to redesignate as described in the waypoint 2 to waypoint 3 section above.

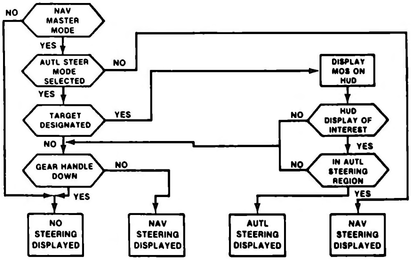


Fig. 19 AUTL Steering Control Logic

The HUD is declared the display of interest at a range of between 1-2 nautical miles from the final approach ground track. The Nav steering azimuth command bar is displayed in the HUD unless several conditions are met as illustrated in Figure 19. These conditions are described below.

First a TDP must be designated and the HUD declared the display of interest. The aircraft needs to be within the display steering region shown in Figure 20. This region extends from a point 6 nautical miles downwind of the TDP.

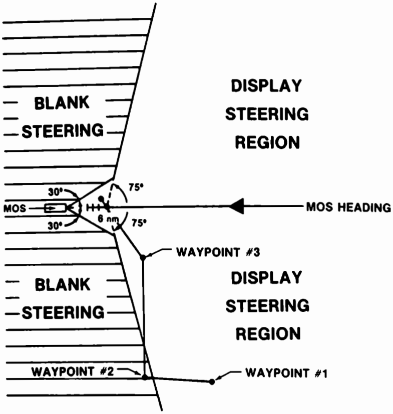


Fig. 20 Steering Display and Blanking Region

The new HUD format is shown in Figure 21. The only detectable change is the steer mode readout changes from NAV to AUTL. The steering guidance algorithms transition from NAV to AUTL Bankline steering. This steering initially directs the aircraft to a point 6 nautical miles downwind of the TDP (point "D" in Figure 12). Then the steering will provide guidance for a turn onto final approach ground track at approximately 1 nm from point "D". The pilot obtains the desired ground track by keeping the velocity vector aligned with the azimuth steering bar.

Ground Track To Touchdown.

The pilot must accomplish the following tasks after the aircraft is on final approach ground track. First, the flaps and landing gear are lowered changing the steering from AUTL Bankline to AUTL Final Approach. This enables final approach elevation and azimuth steering. The elevation steering provides guidance to obtain a 3 degree glideslope. Next, the antiskid system, autobrake system, and the SLAND mode are engaged. The pilot also obtains a landing speed of 119 knots while maintaining glidepath and ground track. Each of the above tasks is described below.

The flaps are deployed as soon as the aircraft is on final approach ground track. Flap deployment puts the IFPC system in the STOL mode of operation; consequently, the HUD readout changes to STOL. This mode utilizes thrust vectoring/ reversing to maximize performance for takeoff and normal approach.

The landing gear is lowered right after flap deployment. This causes several HUD symbology changes. First, the bankline scale, range to steer point, and time to steer point indicators are removed from the HUD to reduce clutter. Then all the alphanumeric readouts are shifted 4 degrees down to place them closer to the over-the-nose view of the TDP. New symbology added to the HUD includes the E-bracket, MOS, vertical velocity readout, radar altimeter, and the azimuth steering bar is replaced by the AUTL steering carets. The gear down HUD format is shown in Figure 10. For further definitions of this HUD symbology refer to the HUD Format Description section of this paper.

It should be mentioned here that the MOS symbol in the HUD would have been displayed as soon as the TDP and MOS heading were designated, and the MOS was in the field-of-view of the HUD. Also, the VANES are illuminated if the pilot commands thrust reversing.

The AUTL steering carets guide the pilot to a 3 degree glideslope along the final approach ground track. While following the steering commands, the antiskid system and autobrake system are engaged. If the autobrake system disengages the AUTOBRK

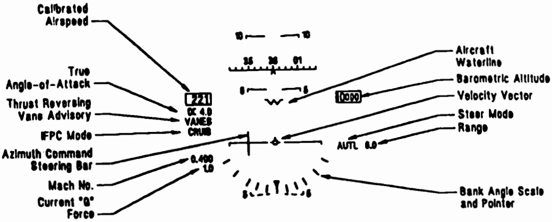


Fig. 21 HUD Display with Gear Up and Bankline Steering

readout in the HUD will appear, flash for 5 seconds, and then turn off.

Approximately 2 nautical miles from the designated TDP the pilot selects SLAND mode. The aircraft airspeed at this point should be approximately 135 knots. The only changes in the HUD symbology are the IFPC mode readout changes to SLAND and the thrust reversing vanes advisory illuminates. In the SLAND mode the IFPC system automatically keeps airspeed constant for a given throttle setting. Before the aircraft reaches an altitude of 200 feet the pilot will obtain an airspeed of 119 knots. He does this by positioning the throttle based on E-bracket and calibrated airspeed information. A complete description of SLAND mode can be found in reference 1.

In this ALG scenario the aircraft does not break out of the clouds until the aircraft altitude is below 200 feet. When this happens, the pilot has his first chance to visually verify the accuracy of the SAR designation. If the pilot is dissatisfied with the designated TDP, he can redesignate by utilizing the HUD's velocity vector. After redesignation, the flight directory steering commands guide the aircraft towards the new TDP.

The final task before touchdown is to flare the aircraft on a 2 degree glideslope. The pilot utilizes the flare cue as described in the HUD Format Description section. The 2 degree flare results in touchdown 30 feet past the designated TDP at a sink rate of 8 feet per second.

Touchdown Through Rollout.

At touchdown, maximum thrust reversing is commanded by pulling the throttle full aft. Upon weight-on-wheels, the autobrake system engages, control laws change to ground handling, and the HUD's IFPC readout changes to STOL. The pilot maintains the desired heading using the control stick and rudder pedals. The ALG task is complete when the aircraft stops.

ALG Evaluations

The version of the ALG system previously described is the result of four evaluations conducted at the FDL simulation facility. The objectives of the evaluations were to analyze and improve the performance of the ALG system. After each test, modifications were made to improve the system. The final version of the ALG system differs from the original as described below. To comprehend the following discussion it is essential for the reader to be familiar with the contents of the ALG Task Description section of this paper.

During engineering studies, deficiencies in the original implementation of the Bankline/Final Approach steering algorithms were discovered. These deficiencies were in gross acquisition of the final approach glidepath and ground track. ALG Test I, II, III, and IV provide a detailed description of the approach used and modifications made to solve these problems.

Figure 12 illustrates the final version of the ALG scenario. This scenario was originally defined with an approach distance of 4 nm instead of 6. The 4 nm distance was selected to minimize APG-70 and INS errors. ALG Test II results indicated a greater approach distance was required to reduce pilot workload.

The original waypoint steering was a modified Bankline algorithm, instead of the NAV steering. This algorithm provided steering cues to obtain the ground path along a line from last to current steer points. These steer points were chosen to optimize radar look angles of the TDP. Keeping the aircraft on this predetermined ground path maximized the performance of the APG-70. ALG Test III results indicated that this level of steering was not required due to increased situational awareness resulting from ALG Test II.

Figure 19 illustrates the final version of the logic used to select Bankline and Final Approach steering. The original logic automatically provided steering guidance towards the TDP when the designation was completed. ALG Test II results and flight tests at McAir indicated that the pilot required greater control on selecting when he received final approach steering. A requirement for the HUD to be declared as the display of interest was added to the logic.

The final version of the EHSI (shown in Figures 11 and 18) incorporates all the waypoints and the MOS ground track symbol. The EHSI was originally implemented with only the current steer point symbol. ALG Test II results indicated that additional situational awareness was required to reduce pilot workload and increase performance.

Following is a complete discussion of each ALG test. First, the purpose of the test along with method of approach is discussed. Then significant results and appropriate system modifications are described.

ALG Test I

The primary objective of ALG Test I was to evaluate the original versus modified versions of the Final Approach Elevation and Bankline steering. Elevation steering was evaluated by initializing the aircraft above the desired glidepath. Bankline steering was evaluated by initializing the aircraft on base leg and having the simulator operator designate an MOS at various ranges from point "D" of Figure 12. Another objective was to evaluate current pilot procedures from base leg to touchdown. It should be mentioned that the APG-70 radar was not used during this testing. The rest of this section will describe test results and appropriate system modifications.

The original elevation steering drives received a flying qualities rating of level 1 for fine tracking of the desired glideslope, as documented in reference 1. However, when these drives were used for gross acquisition, undesired overshoots occurred as illustrated in Figure 22. The approach to solve the problem was to implement a steering limit algorithm while not altering the fine tracking dynamics. This was accomplished by linearly reducing the steer command as the aircraft approached the desired glidepath. The modified limit eliminated the glidepath overshoots as illustrated in Figure 23.

The original azimuth steering drives also received a flying qualities rating of level 1 for fine tracking of the desired ground track, as documented in reference 1. However, when these drives were used in gross acquisition two problems occurred. The reader should first understand that the Bankline and Final Approach azimuth steering are same steering drives, except a higher gain is selected for final approach.

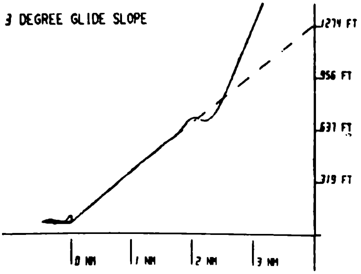


Fig. 22 Original AUTL Elevation Steering

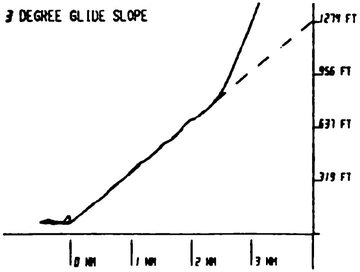


Fig. 23 Modified AUTL Elevation Steering

The first problem occurred on base leg. At a range greater than 2 nm from point "D" shown in Figure 12, the steering commanded the aircraft on a 90 degree intercept to the final approach ground track. This command was considered unacceptable since it resulted in a turn away from the airfield. The solution was to limit the steering command based on the intercept angle between base leg and final approach. Test pilot recommendations were to incorporate this modification into the aircraft's flight control computer.

The second problem occurred during the turn to final approach. The steering drives gave unacceptable bank angle commands resulting in large ground track overshoots as shown in Figure 24. This was a worse case situation since the touchdown point and heading were designated when the aircraft was at point "D". The pilot commented that the original steering was not useable because of the two large overshoots and bank angle commands of 90 degrees. This was corrected, as shown in Figure 25, by reducing the maximum turn rate. The modified steering turned the aircraft onto final approach with no overshoots and a maximum bank angle of 30 degrees.

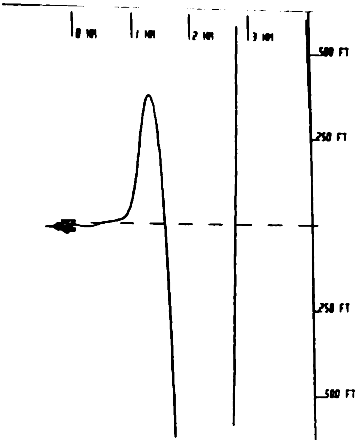


Fig. 24 Original AUTL Azimuth Steering

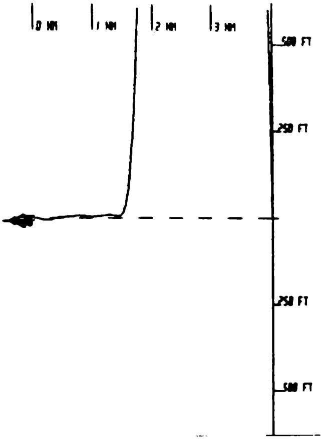


Fig. 25 Modified AUTL Azimuth Steering

Pilot procedures were the last area of interest for ALG Test I. Original procedures stated that the gear and flaps should not be lowered until the aircraft was on final approach.

SUMMARY

The objectives to enhance the ALG system to fully acceptable were successfully completed. Three steering algorithm modifications developed by FDL engineers were implemented in the aircraft's computer. The modifications increased the capability of the aircraft to obtain the desired ground track and glideslope. Recommended display modifications were also implemented to increase situational awareness. Pilot procedures were optimized to reduce pilot workload and several enhancements were made by McAir engineers as the result of flight tests. Several other recommended changes, such as implementing a digital map on the EHSI, would result in further workload reduction and increase safety. FDL testing resulted in the ALG system being fully demonstrated prior to actual ALG flight tests and provided the pilots with confidence in the overall S/MTD ALG design.

Acknowledgements

The authors would like to extend their sincere appreciation to the following people at the FDL simulation facility. Ken Feeser, JoAnn Downey, James Eicher, Jim Lewonski, Mark Sturgell, Brian Stadler, Jeff Slutz, Tom Danube and Mike Denny were all essential in the simulation development and pilot evaluations. We would like to thank Air Force test pilots Lt Col Lewis, Maj Jenschke and Capt Shelly for their dedicated efforts during the four ALG tests. Finally, we would like to thank McAir test pilot Larry Walker and McAir engineers for their support and cooperation during this five year effort.

References

1. Feeser, K., Zeh, J., and Leggett, D. The STOL Maneuver Technology Manned Simulation Test Program, IEEE NAECON Paper 1989.

2. Sturgell, M. and Lewonski, J. Processing Pseudo Synthetic Aperture Radar Images from Terrain Data6000, AIAA Paper AIAA-88-4576-CP.

89-3283-CP

REAL-TIME FLIGHT SIMULATION SUPPORT FOR THE X-31 AIRCRAFT PROGRAM

Steve Zammit and Koos Zwaanenburg
Applied Dynamics International
Ann Arbor, Michigan

Abstract

The real-time simulation of aircraft with many different types of hardware in the loop can be a very challenging task-especially when one has to simulate a high-performance aircraft with fast, nonlinear dynamics, a state-of-the-art flight control system, actuators, and other external hardware simultaneously. This paper discusses in detail the use of the AD 100 simulation computer for the real-time simulation requirements of the X-31 project.

Introduction

The X-31 project is the first international X-aircraft development program in which Rockwell International Corporation's North American Aircraft division is working with Messerschmidt-Bōlkow-Blohm (MBB) of Germany to design and build two Enhanced Fighter Maneuverability (EFM) aircraft for flight testing. This flight testing will focus on controlled flight at very high angles of attack (post-stall maneuvering).

The X-31

The purpose of the X-31 is to prove the feasibility of controlled flight at very high angles of attack, an area where conventional aircraft will stall and fall into a spin. The X-31 will also help to provide a better understanding of the aerodynamics of the high-angle-of-attack flight regime. Instead of losing control at high angles of attack, the X-31 should be able to perform maneuvers that were until now considered impossible, giving it the advantage of being able to point rapidly in a desired direction. To achieve these goals, the X-31 is equipped with flight-control surfaces like rudder, elevons, and canards, and also with paddles in the engine exhaust stream, which can provide thrust vectoring. At low angles of attack, the X-31 will fly like any airplane. At high angles of attack, the stalled flight-control surfaces will not be effective for control, and the thrust-vectoring paddles must take over. For coordinated motion of the aircraft, all flight controls will be governed by a sophisticated fly-by-wire control system, since pilots will not be able to perform complex maneuvers without it.

The X-31 development time is intended to be short, the budget is fairly low. Consequently, off-the-shelf components will be used as much as possible: F404 engine, the cockpit of the F-18, and the landing gear of the F-16.

Purpose of the Simulation

The simulation activities in support of the X-31 project are both real-time and non-real-time in nature. Non-real-time simulation is performed to support the real-time simulations. The real-time models are derived from the non-real-time model, which serves as the "truth model" for all simulation activities. The real-time simulations are necessary for the following reasons: first, for control law and flight-control system verification and validation; second, to verify compliance of the aircraft with handling and flying qualities requirements; and finally, to demonstrate correct performance of the flight-control system in real time.

The real-time simulation activities in support of the X-31 program are broken into two phases. The first is the Real-Time All-Digital Simulation (RTADS) phase. A complete mathematical model of the flight dynamics of the X-31 aircraft, together with the flight-control system, actuators, and control surfaces, is developed and implemented for the purpose of verification of the flight-control laws, pilot evaluation, and handling qualities analysis. In this phase, the simulation program has to communicate with both instruments, stick, and pedals in the mock-up cockpit and a Computer-Generated Image (CGI) system. In the second phase, called Flight Hardware-In-The-Loop Simulation (FHILS), the code for the flight-control computer is removed from the simulation program and replaced by the actual flight-control computer. Furthermore, the aircraft actuators are made external and all system test equipment, the CGI system, and the cockpit instruments and controls are connected to the simulation program. A wide variety of interfacing problems becomes apparent. VMEbus-based systems, 1553-buses, analog hardware for the actuators, and a digital interface to the CGI equipment combine into a complex system that must be orchestrated in a meaningful fashion. The overall result of the FHILS activity will be a full-blown X-31 simulation with hardware in the loop that will be used for overall system evaluation and pilot training.

The fast dynamics of the X-31 aircraft dictate a frame rate of 2 milliseconds or better for the numerical integration of the model state equations. Furthermore, the aerodynamic data base of the X-31 contains several hundred thousand coefficient values, which makes the nonlinear aerodynamic function evaluation task quite challenging with respect to storage requirements as well as computer speed. The flight-control computer is a discrete time system with a sample time of 20 milliseconds. The inclusion of such a system in a simulation of a continuous-time system like the aircraft equations of motion introduces some nontrivial problems. The requirements mentioned above make it clear that a fast and versatile computer system had to be used. The choice of a VAX 11/780[1] and an AD 100 proved to be instrumental in the successful implementation of the RTADS and FHILS phases with the above-mentioned requirements.

SYSTEM 100 Architecture

Applied Dynamics International (ADI) has been involved in solving time-critical simulation of continuous dynamic systems since its founding in 1957. The SYSTEM 100 [4] simulation computer system was introduced by ADI in 1984. It consists of an AD 100, a high-speed, floating-point compute engine; a host controller, a general-purpose digital computer of the VAX family; and ADSIM, a user-friendly simulation language designed specifically for the AD 100. The SYSTEM 100 hardware and software work together to form a complete simulation environment.

The AD 100

The AD 100, a synchronous, bus-oriented multiprocessor compute engine designed for time-critical digital simulation, is the basic fundamental building block of the SYSTEM 100. The AD 100 is a single-user system without an operating system. It is controlled by a multiprocessing VAX host computer running the VMS operating system. Acting as the controller and user interface, the host computer relieves the AD 100 of these interrupt-based tasks. The compute engine needs to be isolated from the overheads and restrictions associated with an operating system in order to achieve and maintain its optimum computation speed or frame rate.

The AD 100 is capable of 20 million floating-point operations per second. The basic system consists of four processors and a host interface tied to a common bus, the PLUSBUS. The PLUSBUS is 105 bits wide, 65 bits of data and 40 bits of address and control. Emitter-Coupled Logic (ECL) devices are used to obtain high computational speed. The four basic processors are the Communication and Control Processor (COM), the

[1] VAX and VMS are registered trademarks of Digital Equipment Corporation

Arithmetic Logic Unit (ALU), the Mulitplier (MUL), and the Storage Processor (STO). Each processor has its own program memory, program counter, and instruction decoder. Each processor has a 64-bit instruction. The COM Processor has a 64K program memory, and the other processors have 4K program memories. Timings in the AD 100 are expressed in terms of a master clock period of 25 nanoseconds. The instruction cycle of the AD 100 consists of four of these phases or periods.

Every arithmetic operation performed on the AD 100 is done in floating-point arithmetic. Calculations are performed using either a 53-bit short-word format or a 65-bit long-word format. Both formats contain 1 sign bit, 12 exponent bits, and either 40 or 53 significand bits. The long-word format is used where additional accuracy is needed, such as in the case of numerical integration to minimize roundoff and truncation errors.

The Storage Processor provides 64K of 65-bit high-speed data storage. Memory accesses and address arithmetic can take place two per instruction cycle. Some simulation tasks such as function generation and memory buffers require large amounts of data storage. It is for these purposes that an optional processor, the Function Memory Unit (FMU), was introduced, which has data memory of 2 million words by 64 bits.

Input/Output Control Processor

The use of hardware-in-the-loop simulation is growing rapidly as a design-and-test environment for the development of products in the aerospace and automotive industries. The interface to hardware in the loop may be either digital as in the case of on-board microprocessors, or may be analog for such hardware as motion-based platforms or system test equipment. Both digital and analog interfaces may be necessary. The Input/Output Control Processor (IOCP) provides both types of interfaces.

The IOCP physically resides in the AD 100 cabinet as a piggyback board to the COM Processor. It does not interface directly to the PLUSBUS. Instead, it passes data through the use of the dual-ported memory of the COM processor. Synchronization between the AD 100 and the IOCP is maintained through the use of a set of common flags.

The IOCP operates independent of the other processors in the AD 100. It has an independent program counter and program memory. The user writes a low-level program, called an ADRIO program, to control it. It is the user's responsibility to synchronize the program running on the AD 100 and the program running on the IOCP. The additional work this requires is more than offset by the added flexibility. The ADRIO program, or IOCP program, and the ADSIM program communicate through the dual-ported memory of the COM Processor. The user is able to set up common buffers between the two programs for passing data to and from the I/O rack to the ADSIM simulation. There are a series of common flags that can be used to synchronize the two programs.

The IOCP is a simple processor capable of seven types of instructions. Each instruction takes 100 nanoseconds to execute. Data transfer instructions are the most widely used of the IOCP's instructions, which include moving data to and from the data buffer through float-to-fix hardware converters, float-to-logic converters, or paths with no conversion. The IOCP is also capable of logic operations on individual bits using its logic processor. The common flags can be manipulated in this fashion. The IOCP is tied directly to the I/O bus in the I/O rack. The bus is 16 data lines, 8 address lines, and 2 control lines. The 8 address lines allow up to 256 individual devices to be addressed in the I/O rack.

ADSIM Environment

With a specific application area in mind, namely real-time simulation, ADI was able to design the AD 100's hardware and ADSIM to handle the necessities of the simulation environment. ADSIM is made up of a high-level simulation language and a run-time interactive control environment.

The ADSIM language is mathematically oriented. Many key elements of a typical simulation are built into the language, such as integration techniques, function generation, and control system nonlinearity functions. A control executive consisting of two programs one that runs on the host controller and one that runs on the AD 100, provides the basis for implementing a model. The control structure built into this executive controls such parameters as system time (simulation time), frame time, and integration step sise. A dynamic section is provided for the model's differential equations. ADSIM allows the model to be implemented as a series of scalar and first-order differential equations. If conditional code is included as part of the model, the control executive takes care of padding the frame such that each frame is consistent with real time. Sorting of the dynamic equations and identifying algebraic loops are examples of some of the capabilites of the ADSIM compiler. Integration is handled using Runge Kutta methods, Adams Bashforth, or Adams Moulton system-level routines. The model development time is much less when a simulation language such as ADSIM is used since many of these standard simulation techniques are built into the language.

ADSIM program development including editing, compiling, and debugging is performed on the host computer. There are two ADSIM compilers: one that produces code to be executed on the AD 100 for time-critical work and one that produces code to be executed on the host computer for non-time-critical experimentation and debugging. The same ADSIM source can be processed by the two different compilers.

Running a program on the AD 100 involves loading the executable code from the host computer into the AD 100 at run time. The user run-time interface consists of a program called INTERACT running on the host computer. INTERACT provides a user-friendly interface for debugging and experimentation, allowing constants to be changed, variables to be displayed, integration methods changed, breakpoints to be set, etc. This environment reduces the time it takes to get the simulation into a state where it can be integrated into the design and testing phase.

X-31 Simulation Implementation

The X-31 simulation implementation can be broken down into four phases or tasks. These tasks are listed as follows

- The Aircraft Dynamic Simulation (ADS)
- The Real-Time All-Digital Simulation (RTADS)
- The Flight Hardware-in-the-Loop Simulation (FHILS)
- The On-Aircraft Flight-Control System Validation

Each of these subtasks of the simulation implementation will now be discussed in detail.

The Aircraft Dynamic Simulation (ADS)

The ADS version of the simulation was the original X-31 aircraft simulation program. The ADS program is a Rockwell NAAO proprietary internal digital simulation program, which may be used as an aid for performing flight-control system synthesis and analysis.

The ADS program is coded in FORTRAN IV on an IBM 370 computer. However, during the development of the X-31 simulation program, a version of ADS was modified by Rockwell personnel to run on the DEC VAX system operating under VMS. The program is designed to facilitate the digital simulation of aircraft and control system dynamics. Six-degree-of-freedom nonlinear equations of motion, angular rate transformation equations, and angles are internally computed in the ADS program.

The ADS simulation program contains a library of routines normally used in control system simulations; this library includes mathematical, switching logic, function generators, limit functions, dead-zone, hysteresis, and component subroutines.

Many different types of systems can be simulated using ADS; however the major use for the program is aircraft simulation. The ADS program allows a user to develop the system model and then link this model into the ADS interactive shell for run-time analysis. This analysis allows. the inspection and modification of system variables and an interactive method of developing simulation time histories.

The first version of the X-31 aircraft was implemented in ADS employing a modular approach to simulation development. The modular approach allowed for a systematic build of the X-31 simulation in both model complexity and fidelity. The results of the "final" version of the non-real-time ADS simulation then served as the "truth-model" for the subsequent real-time all-digital simulation, thus allowing for verification and validation of the RTADS software.

The Real-Time All-Digital Simulation (RTADS)

The RTADS phase of the simulation program is the setup prior to flight hardware delivery and is used for verification and validation of the real-time aircraft, the control laws, and for initial flying qualities evaluation.

The RTADS simulation was to be implemented on the Applied Dynamics AD 100 digital simulation computer using the ADSIM simulation language. Rockwell NAAO personnel determined that internal computer resocrces could not meet the implementation time and iteration rate requirements imposed by the X-31 program time-line schedule and system dynamics, respectively. Thus, the RTADS and FHILS phase work was subcontracted to the Applied Dynamics Applications Department. The time allowed for the RTADS conversion from ADS format to the ADSIM format was five months, and this requirement was easily satisfied. As previously stated, the Rockwell requirement for the numerical integration of the model state equations is 2 milliseconds, and the sample time of the digital flight-control system is 20 milliseconds. The ADSIM model running on the AD 100 executed the equations in 680 microseconds; however, due to the sample time of the digital controller set at 20 milliseconds, an integration step of 1 millisecond was used in order to maintain an integer number of integrations per digital controller sample period. It should also be pointed out that this frame time includes all I/O requirements imposed by the RTADS phase of the simulation. These I/O requirements include analog-to-digital and digital-to-analog converters, and discrete I/O systems which include sense-line and control-line registers, a high-speed digital interface to the DEC VAX system to drive an MBT 1553 bus emulator that in turn drives the cockpit Head-Up-Display (HUD), a high-speed digital interface to the Gould host system for the Evans and Sutherland CGI system to drive the simulation dome graphics, and a high-speed digital interface to the Sun 3 data acquisition system.

The RTADS phase simulated all aircraft systems. These include airframe equations, flight-control computer, actuators, engine, sensors, thrust vector, hinge, landing gear, and ground models. To get a better feel for the overall complexity of the model, a number of specifics are presented below

- Equipment used
  - Applied Dynamics AD 100 with Function Memory Unit (FMU) and analog and digital I/O system
  - Digital Equipment Corporation VAX 11/780
  - Evans and Sutherland CT-6 with Gould Host System
  - mock-up cockpit with flight instruments
  - 36-foot diameter Simulation Dome
  - Sun 3 Data Acquisition System
  - MBT 1553 Bus Emulator
  - three 8-Channel Strip Chart Recorders
  - McFadden Feel System
- Simulation Program Summary
  - 64 state variables
  - 1382 algebraic variables

- 46 generated functions of 1 independent variable
- 29 generated functions of 2 independent variables
- 55 generated functions of 3 independent variables
- 3 generated functions of 4 independent variables
- A total of over 200,000 generated function data points

The RTADS phase also required the ability for a user to interactively trim the X-31 aircraft at an arbitrary set of flight conditions. These included the ability to trim the aircraft both longitudinally and laterally. It was decided to use the ADS non-real-time model to trim the aircraft and to down load the control surface deflections to the AD 100. This required the "splicing" of ADS to the ADSIM run-time environment named INTERACT. Since both ADS and INTERACT are written in FORTRAN, this process, with the help of Rockwell engineers, did not pose a problem. This also provided a cross check between the ADS model and the ADSIM model: if the aircraft is trimmed in one model executing on one system and trimming results are down loaded to a second model executing on a second system, both aircraft should remain in trimmed flight.

The major problems encountered during the conversion process from the ADS FORTRAN to ADSIM were predictable. These problems are inherent in a programming language such as FORTRAN and simply do not manifest themselves in a simulation language such as ADSIM. The problems included improperly ordered dynamic equations in the FORTRAN; ADSIM equations are automatically sorted. This resulted in a correctly ordered FORTRAN code. Another problem was improperly equivalenced FORTRAN common areas; in ADSIM all variables are global, and there is no need for common areas. This prompted us to correct the ADS FORTRAN code. The truth of the matter was that Rockwell NAAO ended up with a better X-31 model than they had before the RTADS project started. Keep in mind however that the ADS model with X-31 code is an extremely large piece of FORTRAN code developed over many years. These comments are not to detract from the utility of or to criticize the X-31 ADS model or the developers; it is simply a fact that a good simulation language will help one to find problems in a model that might otherwise go undetected in a FORTRAN-based simulation.

A diagram depicting the RTADS X-31 real-time simulation hardware configuration is shown in Figure 1.

The Flight Hardware-in-the-Loop Simulation (FHILS)

The FHILS phase of the simulation implementation is an outgrowth of the RTADS phase. This phase of the simulation will be used for the final validation of the flight-control system (FCS) hardware and software in the laboratory before the installation of the FCS and software into the X-31 aircraft. In addition, evaluation of the system design for compliance with flight control, human factors, flight test objectives, and pilot training are also major objectives of the FHILS task.

A piece of equipment developed by Honeywell provides the interfaces between most of the FHILS simulation system. This piece of hardware, named the System Test Console (STC), provides the interface between the FCS, the cockpit, the system test equipment and the real-time simulation of the aircraft dynamics on the AD 100. Malfunction insertion capability for the test of the redundancy management system is also provided. In short, all systems that can be reasonably replaced by actual hardware and software systems are inserted into the FHILS simulation and removed from the ADSIM RTADS simulation code.

The STC is a VMEbus-based system. The interface between the STC and the AD 100 is accomplished via a high-speed digital interface provided by ADI (a dual-ported memory with DEC DR11 protocol) and a VMEbus to DR11-W interface provided by VME Microsystem International Corporation (VMIC). Software protocols developed jointly by Honeywell and Applied Dynamics allow for a high-speed real-time communication between these vastly different computer systems. Data format for the STC is IEEE-754 single precision format, and conversion from the AD 100 internal format to the STC system is performed by the AD 100 in a minimal amount of time.

The Evans and Sutherland with Gould host system is still driven by the AD 100 directly for implementation of the flight dome graphics in the FHILS phase. The AD 100 also communicates with the Sun 3 system for data acquisition and drives the three 8-channel strip chart recorders for the purpose of a real-time monitor of important simulation parameters.

As the implementation of the FHILS task is not yet completed at the time this paper has been written, specifics about the problems encountered, frame times and other areas of interest are not yet available.

A diagram depicting the FHILS X-31 realtime simulation hardware configuration is shown in Figure 2.

The On-Aircraft Flight Control System Validation

This last phase of the simulation program allows for final validation of the Flight Control Computer as installed on the aircraft while sitting on the runway. The System Test Console has been designed with this phase of the simulation in mind, as this piece of equipment can be rolled out onto the runway and interfaced to the on-board FCC. Open and closed loop test of the FCC may be performed via the STC interface. These "strapdown" tests may be performed with and without the engine running. Again, the FCC provides the interface for fault insertion to most of the X-31 on-board computer systems for the purpose of, among other things, the evaluation of the redundancy management systems [3].

Results

The ADS version of the X-31 aircraft simulation was successfully converted to the RTADS simulation by Applied Dynamics personnel in the required time period and with the desired iteration rate requirements. Dynamic fidelity of the X-31 model was increased during the conversion process. A frame time of 680 microseconds was achieved by using the ADSIM simulation language and AD 100 system. Input/output requirements were easily satisfied and verified by Rockwell NAAO personnel.

The work on the FHILS version of the simulation is underway at the time this paper is being written (June 1989). To summarize the FHILS version of the model, one takes the RTADS model and basically eliminates the software controller and actuators. These systems will be replaced by the STC along with the actual hardware (FCS and analog simulation of aircraft actuators). Since we in Ann Arbor do not have the STC and related systems, the following approach has been taken. Using two AD 100 systems, the analogous environment is created: basically using a single AD 100 to simulate the FCS and actuators and the second AD 100 to simulate the airframe dynamics. This has been done in real-time in Ann Arbor and has produced the same results as when the whole system was on a single AD 100. Thus, when we are on-site at Rockwell NAAO, we should need only to plug in the STC and debug the model from there, safe in knowledge that the system has already performed in a predictable manner at least with the second AD 100.

Alternative Implementation Methods

Since most people who work in an engineering environment are not paid to continually reinvent the wheel, it is extremely important that simulation models, functions, and subsystems of a project can be shared by groups of people. ADSIM allows personal libraries, project-related libraries, or even department- and company-wide libraries, which may contain frequently used functions and models. Furthermore, it is possible to set up simulation executives, which are called EXECUTE pairs. A simulation executive is the outer layer of a simulation that takes care of macroscopic phenomena like synchronization of the simulation with the outside world, graphics, starting and stopping the simulation, and the order in which certain combinations of dynamic systems are solved. For example, in a mixed-data system, the continuous-time part will be solved on a frame-by-frame basis, while the discrete-time part will only be solved every T_s seconds. An EXECUTE pair can control the relations between the continuous-time and discrete-time subsystem in a meaningful fashion. In the case of the X-31 simulation, an EXECUTE pair could simplify the interactions between the discrete-time model of the flight-control system and the continuous-time model for the airframe. However, the methodology explained in this section was not used for the X-31 simulation.

An EXECUTE pair is defined as two separate ADSIM blocks; a simulation executive named SIMEXEC and an interactive executive named INTEXEC. The SIMEXEC block is written in ADSIM and defines the order of execution of blocks in the user's ADSIM program. The INTEXEC block is a FORTRAN subroutine which is linked into and called by the INTERACT process. The INTEXEC subroutine allows a user to execute user-written FORTRAN code before and after the AD 100 is started or halted. The INTEXEC subroutine also starts the AD 100 simulation beginning with the control structure as defined in the SIMEXEC block. Since the simulation executives are written in high-level languages, they are readily modified by the user. Three simulation executives are currently available with the ADSIM software. These are:

- An EXECUTE pair for continuous system simulation [1]
- An EXECUTE pair for multiple-frame-rate integration [2]
- An EXECUTE pair for mixed-data system simulation [5]

The EXECUTE pair for continuous system simulation is used in ADSIM programs with a single dynamic block named `DYNAMIC continuous`. The ordinary differential equations representing the system to be simulated are simply placed into the dynamic block. The ADSIM compiler sorts the equations into the proper order to ensure correct evaluation of all quantities. All differential equations are integrated with a single integration step size. For real-time simulation, this integration step size is set equal to the amount of time it takes the AD 100 to execute a single frame. This task is transparent to the user and is performed by the INTERACT task. The general format of ADSIM programs using EXECUTE continuous follows:

```
TITLE {Simulation Title}

DYNAMIC continuous

{ Differential and algebraic equations
      to represent continuous time systems. }

END DYNAMIC continuous

EXECUTE 'continuous'
```

The EXECUTE pair for mixed-data system simulation, named `EXECUTE mixed_data`, is also used in ADSIM programs along with two dynamic blocks. Simulations of this nature are easily partitioned into two or more subsystems. One subsystem consists of algebraic and difference equations that represent the digital or discrete-time control laws. The other subsystem consists of the algebraic and differential equations that describe the behavior of the continuous time system. When creating the control structure for simulation of mixed-data systems, the following factors should be considered:

- Numerical integration of discontinuities introduced by the digital controller residing in the continuous-time subsystem
- Selection of numerical integration methods
- Adjustment of either the sample period or integration step size

- Representation of computational delay in the digital subsystem

The first point to take into consideration is the selection of the integration method. Single-step predictor-type methods that use the current and past derivatives to predict the new state values do so under the presumption of continuous derivatives. This presumption does not hold when simulating mixed-data systems. Strong discontinuites are introduced by the control command as determined by the discrete time system. Self-starting methods, such as the classical, multistep Runge Kutta second-, third- and fourth-order variety, do not rely on this assumption, and thus these methods will be used. However, the classical Runge Kutta methods are not suited for real-time simulation as these methods require external inputs to the integration method before they become available in time. At Applied Dynamics we have anticipated this problem and developed our own Runge Kutta real-time methods with similar accuracy and stability characteristics as the traditional versions. See [1] and [5] for details.

Another consideration is the correct simulation of the digital controllers computational delay. Actual microprocessor hardware does not produce control action to the continuous system instantaneously. The most common method is to compute the control effort and output as soon as the results become available. However, when simulating delay time, the lag must be an integer multiple of the continuous-system integration step size. If you have an estimate of the computational delay for your particular piece of hardware, **EXECUTE mixed_data** will allow you to simulate the effect of both fixed and time-varying computational delays on your continuous system model. Figure 3 shows the relation of delay time to the sample time.

The general format for ADSIM programs using the control structure of **EXECUTE mixed_data** follows:

```
TITLE {Simulation Title}

DYNAMIC discrete

{ Difference and algebraic equations
          to represent digital controller. }

END DYNAMIC discrete

DYNAMIC continuous

{ Differential and algebraic equations
      to represent continuous-time plant. }

END DYNAMIC continuous

EXECUTE 'mixed_data'
```

Figure 4 shows the control structure provided by **EXECUTE mixed_data**.

Let us consider an example of a mixed-data system simulation using ADSIM and **EXECUTE mixed_data**. The system simulated is a simple second-order closed-loop system with a digital PID type controller. Errors introduced due to the digital-to-analog (D/A) and analog-to-digital (A/D) converters are also modelled by invoking a standard ADSIM model named quantizer [1]. Scaling of the input and output quantities is also performed, as would be needed if actual controller hardware were to be used. The system simulated is shown in Figure 5.

The digital controller transfer function, $D(z)$, is defined using z-transform notation and has the following form:

$$D(z) = \frac{M(z)}{E(z)} = \frac{a_1 + a_2 z^{-1} + a_3 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \qquad (1)$$

The continuous-time transfer function, $H(s)$, is defined using Laplace transform notation and has the following form:

$$H(s) = \frac{X(s)}{M(s)} = \frac{K_g}{(\tau_1 s + 1)(\tau_2 s + 1)} \qquad (2)$$

The ADSIM language supports both continuous-time derivatives and discrete-time difference equations. Therefore, the above defined transfer functions, $D(z)$ and $H(s)$, are simply converted to ADSIM notation and placed into the discrete and continuous dynamic blocks, respectively. Note that the transfer function for a zero-order hold is not represented in either the block diagram or in the ADSIM program. The reason for this is that the simulation control structure as defined by the SIMEXEC block in the **EXECUTE mixed_data** pair supplies this effect implicitly. The following program fragment displays the use and the invocation of the EXECUTE pair by the **EXECUTE mixed_data** statement.

```
TITLE "Plant & digital controller"

(* Digital controller difference equations *)

dynamic discrete

(* Generate reference position *)

 xref = test_signal(s_time,system_time)

(* Scale and quantize input, x *)

 xs = kadc*x
 xq = quantizer(a,b,xs)
 xe = ikadc*xq

(* Generate error signal and sequence *)

 e#   = xref-xe
 epf# = e

(* Generate control signal and sequence *)

 m#   = b1*m+b2*mpf+a1*e#+a2*e+a3*epf
 mpf# = m

end dynamic discrete

(* Second order plant differential equations
*)

dynamic continuous
```

```
method  rkrt2

(* Scale and quantize controller output, m *)

 ms = kdac*m
 mq = quantizer(a,b,ms)
 me = ikdac*mq

(* Second order plant equations *)

 x' = y
 y' = 1.0/(tau1*tau2)*(kg*me-(tau1+tau2)*y-x)

end dynamic continuous

(* Specify default data set *)

data tau1 = 4.0, ...

(* Specify controller constants *)

data b1 = 0.540, b2 = 0.460, ...

(* Specify simulation run specifications *)

runspecs endtime    = 10.0,
         sampletime = 1.0,
         frametime  = 50.0e-6,
         delaytime  = 10.0e-3,
         speedup    = 1.0

(* Execute pair for mixed-data simulation *)

execute "mixed_data"
```

Conclusions.

The performance numbers of the AD 100 show that it is possible to implement a high-performance aircraft like the X-31 in a real-time hardware-in-the-loop simulation without the problems associated with an implementation on a general purpose digital computer using FORTRAN. It has also been shown that the use of specialized EXECUTE pairs can simplify the implementation effort of systems with digital controllers governing continuous time systems.

References

[1] *ADSIM Reference Manual Version 6.1.* ADI, Ann Arbor, MI, 1989.

[2] Haraldsdottir, A., and Howe, R.M. *Multiple Frame Rate Integration.* AIAA Flight Simulation Technologies Conference, September 1988.

[3] Snyder, G. *The X-31-A Aircraft Flight Hardware-in-the-Loop Simulation.* Applied Dynamics International Users Society Conference, Keystone, CO, June 1989.

[4] Wright, M. *System 100 Simulation Computer Architecture.* European Simulation Multiconference, June 1989.

[5] Zammit, S., and Zwaanenburg K. *Simulation of Mixed-Data Systems Using the Applied Dynamics Simulation Language ADSIM.* European Simulation Multiconference, June 1989.

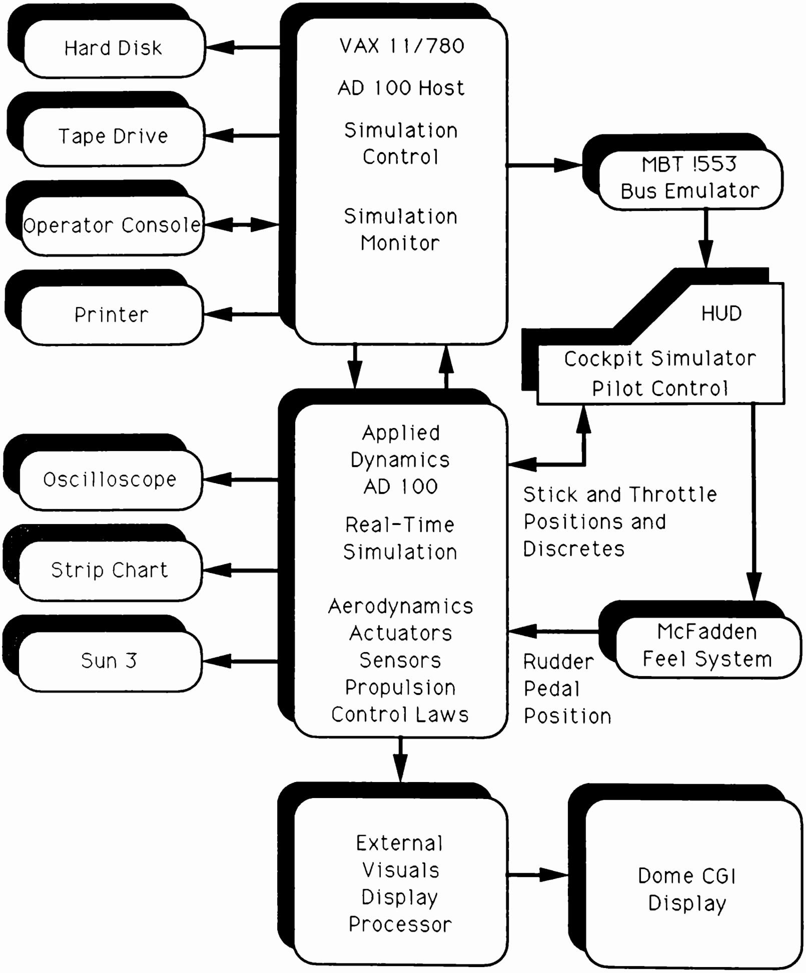


Figure 1: RTADS simulation hardware configuration.

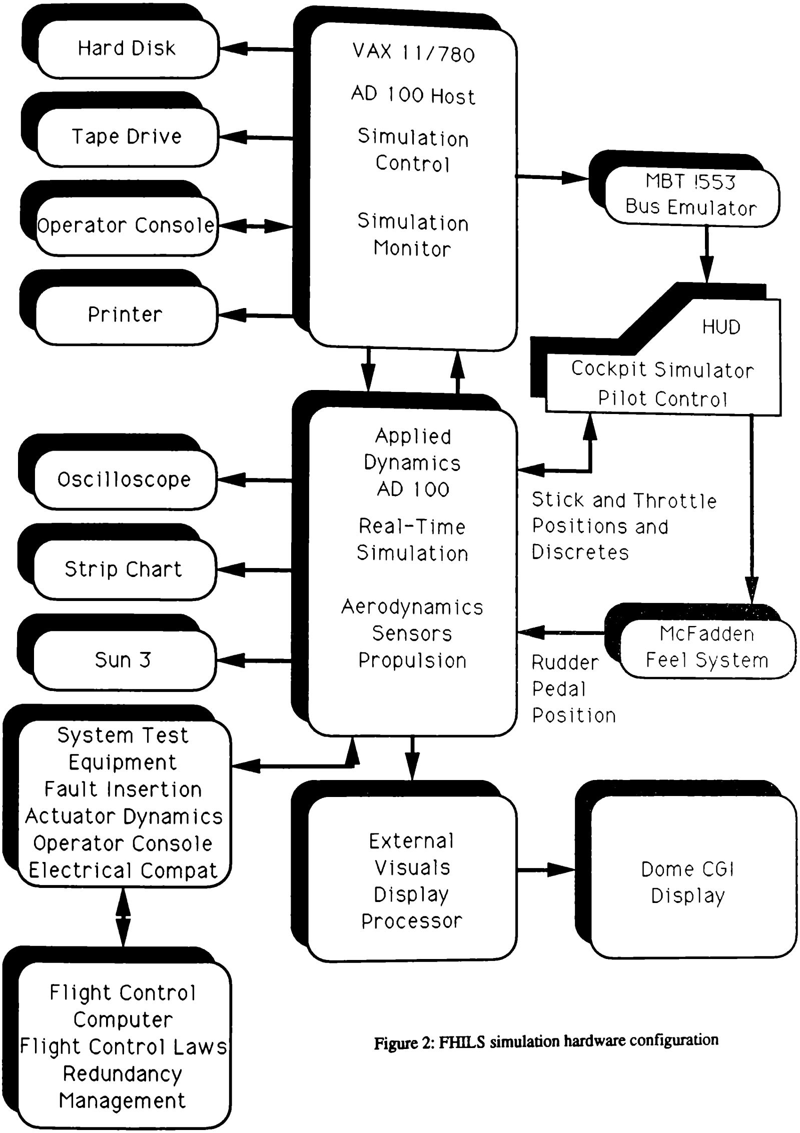


Figure 2: FHILS simulation hardware configuration

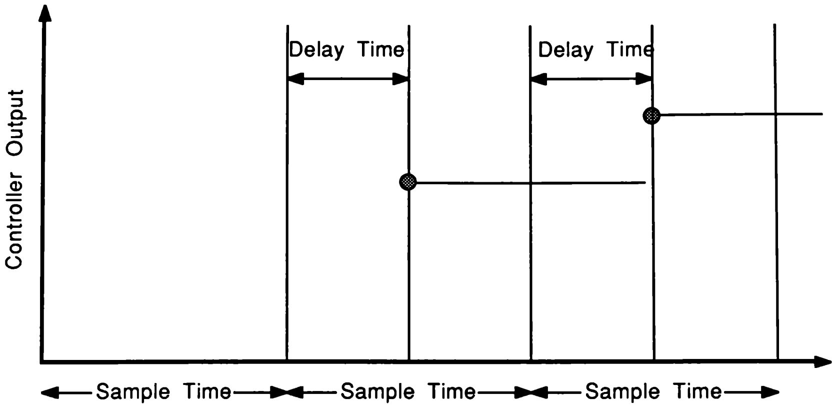


Figure 3: Relation of delay time and sample time.

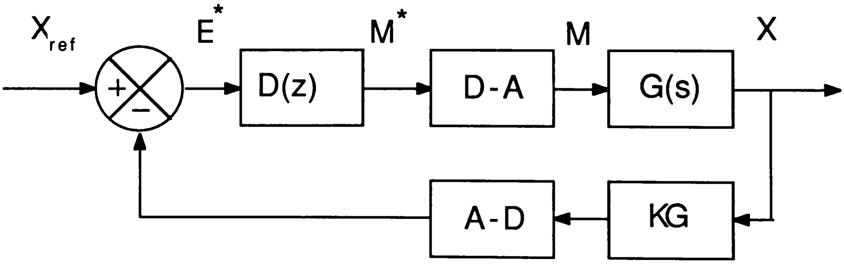


Figure 5: Mixed-data system.

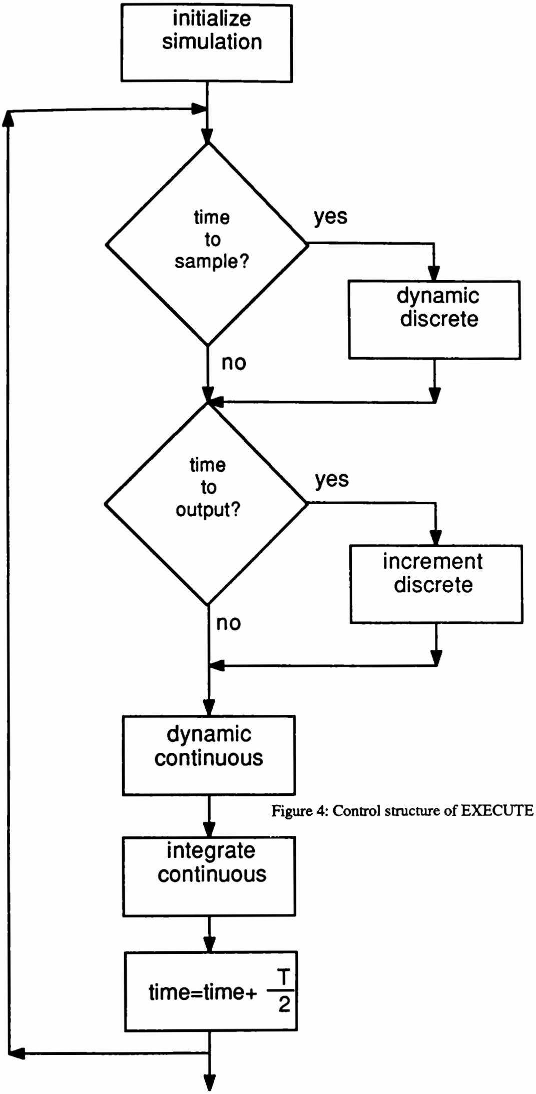


Figure 4: Control structure of EXECUTE mixed-data.

89-3284-CP

MODELLING ATMOSPHERIC EFFECTS FOR FLIGHT SIMULATORS

Paul A. Robinson*

University of Toronto Institute for Aerospace Studies, Toronto, Ontario, Canada

Abstract

This study investigates the degree of complexity required in the simulation of turbulence and thunderstorm downbursts for flight simulation. A turbulence model is presented which contains all the correlations found in homogeneous isotropic turbulence. Several simplifications and alternate models are considered. This paper also presents further developments in the simulation of thunderstorm downbursts. A single ring vortex system and a more complicated triple ring vortex system are compared to Joint Airport Weather Studies (JAWS) data sets. By means of pilot evaluations on the UTIAS B-747 Flight Research Simulator, it was found that the inclusion of the isotropic turbulence correlations did not seem to affect the pilot and not much difference was perceived among the turbulence models. It was also found that the inclusion of the gust time derivatives in the aircraft equations of motion was an important factor affecting the realism of the simulation. Downburst evaluations showed that both single and triple ring vortex systems were able to simulate actual downbursts (JAWS data). It is suggested that the ring vortex models be expanded upon to include more than one cell.

Nomenclature

$\underset{\sim}{A}$	force vector.
$f(r)$	longitudinal turbulence correlation.
FPDP	Flight Path Deterioration Parameter.
$g(r)$	lateral turbulence correlation.
GS_+	on or above glideslope FPDP. $= \frac{1}{T_{GS_+}} \int_0^{T_{GS_+}} \frac{H}{H_{G/S}} dt$
GS_-	below glideslope FPDP. $= \frac{1}{T_{GS_-}} \int_0^{T_{GS_-}} \frac{H}{H_{G/S}} dt$
H	aircraft c.g. height above ground.
$H_{G/S}$	glideslope height above ground.
j	$\sqrt{-1}$
k	$\sqrt{k_1^2 + k_2^2 + k_3^2}$
$\underline{k}$	wave number expressed in cycles per metre: $[k_1\ k_2\ k_3]^T$.
l_h	distance between aircraft c.g. and stabilizer.
$\underline{n}(\underline{x})$	three independent Gaussian white noise sources: $[n_1\ n_2\ n_3]^T$.
$\underline{N}(\underline{k})$	the Fourier transform of $\underline{n}(\underline{x})$.
r	$\sqrt{r_1^2 + r_2^2 + r_3^2}$.
r_o	reference ring vortex radius.
$\underline{r}$	spatial separation between two points: $[r_1\ r_2\ r_3]^T$.
$R_{xx}(\tau)$	the autocorrelation function for $x(t)$ at time delay τ.
T_{GS_+}	time spent on or above glideslope.
T_{GS_-}	time spent below glideslope.
T_{V_+}	time spent at or above approach speed.
T_{V_-}	time spent below approach speed.
V	airspeed.
$\underset{\sim}{V}$	airspeed vector: $[u_B\ \ v_B\ \ w_B]^T$.
V_A	approach airspeed (149 kts).
V_+	on or above approach speed FPDP. $= \frac{1}{T_{V_+}} \int_0^{T_{V_+}} (V - V_A)\, dt$
V_-	below approach speed FPDP. $= \frac{1}{T_{V_-}} \int_0^{T_{V_-}} (V - V_A)\, dt$
$\underset{\sim}{W}$	wind velocity vector: $[W_x\ W_y\ W_z]^T$
δ_{ij}	Kronecker delta.
$\sigma(t)$	modulating variable.
σ^2	turbulence intensity.
$\underline{\Phi}(\underline{k})$	turbulence spectrum function (3×3 matrix with elements $\Phi_{ij}(\underline{k})$).
Γ_{ref}	reference ring vortex circulation strength.

Notation

$\underline{b}$	column matrix.
$\underline{B}$	matrix.
$\underline{B}^H$	the Hermitian transpose of $\underline{B}$ (the complex conjugate of $\underline{B}^T$).

Subscripts

B	body-axes frame of reference.
I	inertial frame of reference.

Introduction

The faithful simulation of the atmospheric environment in commercial flight simulators is an important factor in establishing the realism of the simulation. Low-altitude turbulence, while seldom a serious threat to an aircraft's safety in itself,increases the pilot workload and may degrade the pilot's capability to deal with emergency situations. Many turbulence models have been proposed in the past and used with varying degrees of success on flight simulators. The effect of spatially correlated gusts, and correlated gust components on a flight-simulation has been of interest for some time [1]. To date it is not clear whether or not they have an effect on the simulation.

On the other hand, the wind shears produced by thunderstorm downbursts have been the cause of several catastrophic accidents in the past. Whilst the best strategy for thunderstorm downburst encounters is avoidance, pilots must be trained for the situation of an inadvertant encounter. Training for this type of wind shear encounter can only take place in a flight simulator [2]. It is therefore important to evaluate the capability of current wind shear models to simulate actual wind shear occurrences.

*Graduate Student, Member AIAA.

Turbulence Models

Three-Dimensional Correlated and Uncorrelated Models

From the theory of isotropic turbulence, the three-dimensional correlation tensor is [4]

$$\frac{R_{ij}(\underline{r})}{\sigma^2} = [f(r) - g(r)]\frac{r_i r_j}{r^2} + g(r)\delta_{ij} \qquad (1)$$

The Fourier transform of the correlation tensor yields the spectral tensor

$$\Phi_{ij}(\underline{k}) = \frac{E(k)}{4\pi k^4}\left(k^2\delta_{ij} - k_i k_j\right) \qquad (2)$$

where $E(k)$ is the von Kármán energy function [5]

$$E(k) = \frac{110}{9}\sigma^2 L \frac{(2\pi a L k)^4}{[1 + (2\pi a L k)^2]^{(17/6)}} \qquad (3)$$

Three turbulence velocity components can be generated by

$$\underline{U}(\underline{k}) = \underline{G}(\underline{k})\ \underline{N}(\underline{k}) \qquad (4)$$

where $\underline{U}(\underline{k})$ and $\underline{N}(\underline{k})$ are the Fourier transforms of the turbulence and white noise respectively, and the transfer function $\underline{G}(\underline{k})$ comes from the solution to [6]

$$\underline{\Phi}(\underline{k}) = \underline{G}(\underline{k})\ \underline{G}^H(\underline{k}) \qquad (5)$$

The inverse Fourier transform of $\underline{U}(\underline{k})$ produces the spatial turbulence velocity field $\underline{u}(\underline{x})$. This model assumes Taylor's hypothesis of a frozen flow field [5].

There are two possible formulations of the transfer function matrix $\underline{G}(\underline{k})$. If $\underline{G}(\underline{k})$ is taken to be

$$\underline{G} = \begin{bmatrix} G_{11} & 0 & 0 \\ G_{21} & G_{22} & 0 \\ G_{31} & G_{32} & G_{33} \end{bmatrix} \qquad (6)$$

the resulting three components of the velocity field $\underline{u}(\underline{x})$ will be correlated with each other according to Equation 1. Alternatively, $\underline{G}(\underline{k})$ can be taken as

$$\underline{G} = \begin{bmatrix} G_{11} & 0 & 0 \\ 0 & G_{22} & 0 \\ 0 & 0 & G_{33} \end{bmatrix} \qquad (7)$$

This results in the three components being independent of each other. The latter is the formulation employed in Reference [7].

The above procedure produces two separate models. Throughout this work, the turbulence produced by the formulation in Equation 6 will be referred to as the 3-D correlated turbulence model, and that produced by Equation 7 as the 3-D uncorrelated model. It should be borne in mind however, that both models contain the spatial correlations within each component as found in isotropic turbulence.

As the aircraft flies through this field, the turbulence velocity components at its center of gravity are determined and transformed from the ground-fixed $\mathbf{F}_I$ frame into the aircraft body frame $\mathbf{F}_B$. In a similar manner the significant turbulence velocity gradients are computed in $\mathbf{F}_B$ as well. These gradients and the associated rotational motion are given by [8]

$$p = \frac{\partial W_z}{\partial y} \qquad q = -\frac{\partial W_z}{\partial x} \qquad r_1 = -\frac{\partial W_x}{\partial y} \qquad r_2 = \frac{\partial W_y}{\partial x}$$

This generates turbulence time series consisting of three velocities and three gradients which are used to produce aircraft forces and moments.

It is generally accepted that turbulence is a non-Gaussian process [9]. The amplitude distribution tends to be patchy with significant periods of low and high intensity. The method of producing turbulence was based on the filtering of Gaussian white noise, and as a consequence, the initial turbulence field was also Gaussian. In order to increase the realism in flight simulation applications, it is necessary to alter the statistical properties of this initial turbulence while maintaining the spectral characteristics. A suitable process has been introduced in Reference [10], and is given by

$$w(t) = \sigma(t)\ z(t) \qquad (8)$$

where it is desired to have the spectral properties of some time series $w(t)$ identical to those of $z(t)$ while at the same time having a different probability density function. This is achieved by employing a stochastically fluctuating $\sigma(t)$. It can be shown that the desired properties can be achieved when

$$\frac{d^2 R_{\sigma\sigma}(\tau)}{d\tau^2}$$

is suitably small as $\tau \rightarrow 0$. This technique has been used to process the initial turbulence field in order to generate patchy turbulence. At each time step a new value of $\sigma(t)$ is calculated, and this value is used to modulate the turbulence gusts u_G, v_G, w_G, p_G, q_G, and r_G.

Modulated Turbulence Model

The above models required the turbulence to be generated in the frequency domain. Alternatively, by filtering white noise using simple linear filters, turbulence may be generated in the time domain. In the past, much attention has been focused on the three-filter turbulence generation models proposed in [9], [11], and [12]. The Modulated Turbulence Model is an adaptation of the three-filter model, but instead of using three filters to produce a patchy time series, the modulation process described above (Equation 8) is used, reducing the number of filters required for each gust series to one. The u_G, v_G and w_G gusts are generated with the Dryden spectral forms [13], and rolling and yawing gusts are generated using filters with transfer functions based on Reference [12]. Pitch gusts are produced by applying a time delay to the α_G at the aircraft centre of gravity and applying it to the stabilizer, i.e.

$$\alpha_{G_{stab.}}(t) = \alpha_{G_{c.g.}}\left(t - \frac{l_h}{V}\right) \qquad (9)$$

All the linear filters are driven by Gaussian white noise sources. These sources are statistically independant, the u_G and r_G filters sharing the same source, and the w_G and p_G filters sharing their own source. All gusts, u_G, v_G, w_G, p_G and r_G, are multiplied by a suitable stochastic $\sigma(t)$ function to generate the required statistical properties. All gust components use the same value of $\sigma(t)$.

This model differs from the pair of three-dimensional models in that the Dryden spectral form is employed. Also the turbulence from this model does not contain the spatial correlations, or correlations among the components found in the first two models.

Statistical Discrete Gust (SDG) Method

As a fourth model, a very different approach to generating turbulence is employed. Developed at the Royal Aircraft Establishment, the SDG method uses as a basic element, a ramp gust followed by an exponential washout ([14], [15], [16]). The exact procedure for generating turbulence series using this model is given in Reference [14]. The u_G, v_G and w_G gusts are generated at the aircraft's c.g. as in the previous models. Roll and yaw gusts are calculated by generating u_G and w_G gusts independently at each wingtip, and dividing the difference by the separation distance. Pitch gusts were generated using the time delay process of Equation 9.

The above four turbulence models represent three very different methods of generating turbulence for flight simulators. The models and their properties are summarized in Table 1. The variation of turbulence intensities and lengthscales with height was according to the data in Reference [17]. In order to generate the patchy nature of turbulence, the fourth order central moment of all the gusts series was set to give a kurtosis (the ratio of the fourth-order moment to the square of the second-order moment) of 4.5. In the SDG model, the F value was set at 0.4 [14]. During the turbulence evaluations detailed below, the surface wind speed (at 10 m height) was 055° at 20 knots (the runway heading is 055°). The mean wind variation with height was according to the power law given in [17].

By means of pilot evaluations, the realism of each model and its value in a piloted flight simulation may be examined.

Thunderstorm Downburst Modelling

In recent years increasing emphasis has been placed on training airline pilots for flight through wind shear conditions. Several downburst models for flight simulators have been proposed in the past using elements of potential flow theory to simulate the velocity field ([18], [19], and [21]). While it appears that such models can produce aircraft response similar to that observed in real encounters, such models must be made to emulate actual downburst conditions as closely as possible. Such a comparison could not be made until recently when results from the Joint Airport Weather Studies (JAWS) were made available for simulation applications [22]. This allowed direct comparison between the simpler models and the larger JAWS data sets.

In Reference [19] it is suggested that a downburst model using a single ring vortex (and its image) will not produce a realistic wind field; in particular the ratio of the maximum vertical wind speed to the maximum horizontal wind field would be smaller in the single ring case. It is suggested that several concentric ring vortices of different strengths be employed to obtain a more realistic profile [20]. Strengths would increase toward the centre to produce a more intense, localised downdraught. Two models were studied in this work. The first employs a single ring vortex, and the second uses three concentric ring vortices with a strength distribution given by

$$\Gamma_i = \Gamma_{ref}\left(1 - \sin\left(\frac{\pi}{2}\frac{r_i}{r_o}\right)\right) \qquad i = 1,2,3 \qquad (10)$$

The models were implemented in such a way that both required equal computation time in the simulation, thereby removing a common argument against the more complicated models in the past.

Two data sets were selected from the JAWS data in Reference [22]; corridors $\overline{AB}$ and $\overline{CD}$ from data set 5AU1847. A vertical cross- section of these downbursts are shown in Figure 1 (from [22]), as well as the orientation of the ILS 3° glideslope used in these evaluations. These were chosen as they had already been classified as challenging, and consisted of one well-defined downburst, but having different vertical wind structures. The single and triple ring models' parameters were adjusted in such a way that both generated wind components along the glideslope as close as possible to those generated by the JAWS data sets. This gave rise to six separate wind shear sets. Since there is no turbulence information in the JAWS data sets, none of the wind shears contained any turbulence.

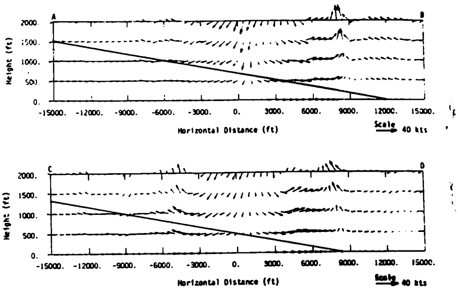


Figure 1: JAWS Corridors $\overline{AB}$ and $\overline{CD}$ [22]

In this work the JAWS data models will be referred to using the prefix J, and the single and triple ring models by the prefixes SR and TR respectively. References to JAWS corridor data sets $\overline{AB}$ and $\overline{CD}$ use the suffices 1 and 2 respectively; e.g. the single ring vortex model simulating the JAWS corridor $\overline{CD}$ is model SR2 (see Table 1).

Experimental Design

All the above models were implemented on the UTIAS Flight Research Simulator shown in Figure 2. The aircraft simulated was a Boeing 747. Details of the simulation can be found in References [23], [24], and [25]. The simulator was upgraded from that described in Reference [23] to include a three-window colour visual display, and a hydraulic control loading system.

In order to evaluate the turbulence and downburst models, use was made of current airline pilots. All pilots had experience in training simulators. In all twelve different pilots were

Table 1: Turbulence and Downburst Models

Model Number	Attributes
M1	Modulated turbulence model.
M2	Statistical Discrete Gust model.
M3	3-D spatially correlated model; no correlation among gust components.
M4	3-D spatially correlated model; gust components correlated with each other.
NT	No turbulence
J1	JAWS $\overline{AB}$ data set
SR1	Single ring vortex model simulating JAWS $\overline{AB}$ data set
TR1	Triple ring vortex model simulating JAWS $\overline{AB}$ data set
J2	JAWS $\overline{CD}$ data set
SR2	Single ring vortex model simulating JAWS $\overline{CD}$ data set
TR2	Triple ring vortex model simulating JAWS $\overline{CD}$ data set
NS	No wind shear

employed, six for the turbulence evaluations, and six for the downbursts tests. Their backgrounds and experience are listed in Table 2. Although only one had Boeing 747 experience, with sufficient training useful results could be obtained from all of them. There was also the added advantage of capturing a broad cross-section of the commercial pilot community.

Table 2: Pilot Experience

(a): Turbulence Evaluation Pilots

Subject number	Current position	Total flying hours	Transport flying hours	Flight simulator hours
1	F/O*, L-1011	15000	9000	400
2	F/O, DC-9	2000	800	50
3	F/O, B-767	9000	7500	350
4	F/O, DC-9	9500	7000	1500
5	Sim. Instructor, DC-9	17000	12000	300
6	F/O, B-767	9500	8500	275

*First Officer

(b): Downburst Evaluation Pilots

Subject number	Current position	Total flying hours	Transport flying hours	Flight simulator hours
1	Pilot, DASH-8	9000	7000	200
2	Pilot, DASH-8	8300	800	200
3	F/O, DC-9	9000	8000	250
4	Captain, DC-9	10000	8000	200
5	Captain, B-767	30000	25000	1000
6	Development Pilot	9000	8600	1000

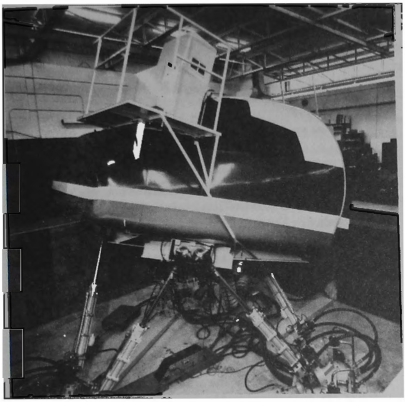


Figure 2: UTIAS Flight Research Simulator

Turbulence Evaluations

The task and analysis used in the turbulence evaluations is very similar to that used in Reference [23]. The primary task in the turbulence evaluation is shown in Figure 3. It was selected to represent the terminal portion of a flight and comprises the following zones:

1. Heading and altitude hold.
2. VOR radial intercept.
3. VOR radial track.
4. Deceleration while tracking VOR radial.
5. Descent.
6. Sidestep manoeuvre to capture the ILS.
7. ILS approach to touchdown.

In both the turbulence and downburst evaluations, the pilots were instructed to judge the quality of the atmospheric model and no other aspect of the simulation. It was also made clear that they should rate the models relative to that which would be experienced in the actual aircraft, and not in other simulators. The various models and their attributes are summarized in Table 1.

Both subjective and objective measurements were used to evaluate the effects of the turbulence models. The subjective ratings used the scale shown in Figure 4. The rating scale is based on that reported in References [23] and [26] and the adjectives on the left-hand side are spaced so as to produce an equal interval scale. In addition, the pilots were encouraged to add any comments they wished. Immediately after each trial, the pilots were asked to mark on each vertical line their assessment of the turbulence. The pilot flew the full task (Figure 3) for each turbulence model (in a randomized order), and once with no turbulence. In addition to this, the pilots flew paired comparison tests as reported in Reference [23]. This allowed the pilots to compare the turbulence models. The first set of paired comparison tests was over zones 3, 4, and 5, and the second over zone 7. Each takes about 4 minutes to fly. A single trial consists of flying the particular task twice in close succession but with different turbulence models. After flying each pair, the pilot was asked to indicate which model was more realistic. All possible pairs were presented to the pilot using a randomized Latin-Square design. The analysis is described in Reference [27].

Downburst Evaluations

The downburst evaluations required the pilots to fly an approach to land (zone 7). They were asked to fly the approach to the best of their ability, and go-around only when a landing seemed impossible. Go-around procedures were at the discretion of the pilot and his previous training, but most agreed that the approved procedure was to apply full throttle, pitch up to the stick-shaker warning, and leave the aircraft configuration alone until clear of the perceived danger. In the event of a landing, brakes were applied to a full stop. During the tests they were given no instruction in how to fly the various shears.

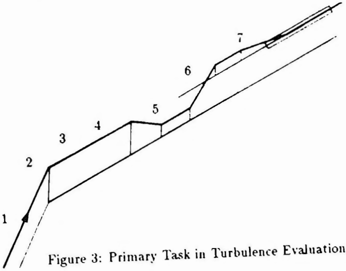


Figure 3: Primary Task in Turbulence Evaluations

Table 3: Analysis of Variance for Complete Set of Results for Turbulence Evaluations

Effect[a]	Degrees of freedom	Sum of squares	F value	$P(x > F)$[b]
Subjects	5	48.4007	14.7027	<0.0001 (<0.0001)
Treatments	3	0.6325	0.3202	0.8107 (0.0012)
Subjects × Treatments	15	32.4878	3.2896	0.0001 (<0.0001)
Variables	9	47.3442	7.9899	<0.0001 (<0.0001)
Subjects × Variables	45	84.0605	2.8372	<0.0001 (<0.0001)
Treatments × Variables	27	9.9196	0.5580	0.9606 (0.7353)
Residual	135	88.8827		
Total	239	311.7280		

[a] "Treatments" refers to the 4 turbulence models, and "Variables" refers to the rating parameters
[b] Values in parentheses are calculated including the NT data

Table 4: Analysis of Variance Summary for Turbulence Evaluations; $P(x > F)$

Parameter	Subjects	Treatments
Overall impression	0.3050	0.6627
Variation of the turbulence parameters with height	0.0729	0.6846
Roll	0.0219	0.9570
Pitch	0.0062	0.4456
Yaw	<0.0001	0.4609
Vertical acceleration	0.0007	0.9797
Relation between the gust components	0.3075	0.9017
Heading control	<0.0001	0.1065
Speed control	0.3312	0.8657
Altitude control	0.9351	0.7388

Results and Analysis

Turbulence Evaluations

Pilot Ratings The results of an analysis of variance carried out on the pilots' ratings from the scales in Figure 4, show most significant effects were subject and variables (rating parameters). If the analysis of variance included the data for the runs without turbulence, the treatment effects became significant. The results are summarized in Table 3. They suggest that the pilots' ratings were affected by the turbulence but there was not much difference between the models. In order to investigate the effects of the individual rating parameters, an analysis of variance was performed on each item separately. The results (excluding the NT data) are shown in Table 4. Repeating the analysis including the NT data produced essentially the same results. It can be seen that subject effects are most significant with heading control being the most significant parameter for treatment effects.

Paired Comparison Tests In these tests, pilots were asked to compare turbulence models over short flight segments. The results are shown in Table 5. From the analysis, pilots rated the models in the same order for both segments. However the within-pilot consistency [27], k, was low for some pilots. Overall the within-pilot consistency was better on the approach segment than when tracking the VOR radial. The results were recomputed excluding the results from pilots producing zero consistency ($k = 0$) in both segments, and the ordering remained the same. As a consequence of the poor within-pilot consistency, the inter-pilot consistency, W, was also low for both segments.

From the previous results it may be concluded that the pilots cannot discern between turbulence models in their subjective ratings. It now remains to investigate the aircraft and simulator response, and pilot performance.

Pilot Performance The pilots' control activity and performance were studied in an effort to determine how the different turbulence models affect the aircraft. Also an analysis of variance was carried out on each of the parameters studied. From these results several observations can be made. Firstly there is no well-defined difference in the aircraft response or pilot performance to the different turbulence models. It was found that the method of implementing pitch gusts using the α time delay in Equation 9 produced large pitch deviations in models M1 and M2. Such motions were found to be unrealistic and a major source of altitude departures. It was found that in the early part of the task (up to the descent) M2 produced the largest standard deviation of control inputs, with M1 close behind. The was largely due to the pitch deviations.

Table 5: Summary of Turbulence Evaluation Paired Comparison Tests

Segment	Rating (best → worst)	k	W
1	M4 – M3 – M2 – M1	(.5,.5,.5,1.,1.,0.)	0.200
2	M4 – M3 – M2 – M1	(1.,1.,1.,.5,0.,.5)	0.133

On the approach where the pilot is flying the ILS with close airspeed control, M3 and M4 produced the largest deviations in airspeed (and consequently throttle inputs), and glideslope error. This can be traced to the method of implementing the turbulence in the equations of motion. In vector form, the translational acceleration equations are given by

$$m\dot{\underline{V}}_I = \underline{A} + m\underline{g} \tag{11}$$

also

$$\begin{aligned} \underline{V}_I &= \underline{V} + \underline{W} && (12)\\ \Rightarrow \dot{\underline{V}}_I &= \dot{\underline{V}} + \dot{\underline{W}} && (13) \end{aligned}$$

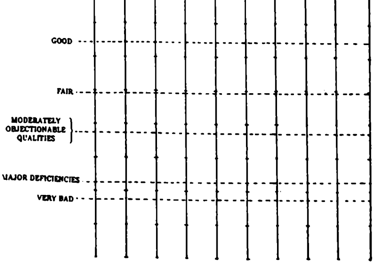


Figure 4: Pilot Rating Scales

Substituting 13 into 11 gives

$$m\dot{\underset{\sim}{V}} = \underset{\sim}{A} + m\underset{\sim}{g} - m\dot{\underset{\sim}{W}} \tag{14}$$

The implementation of models M1 and M2 do not include the time derivative of the wind vector on the right-hand side. This was so because it was the way they have been implemented in the literature ([12] and [21]), also these models do not provide a direct source of these terms. Although these time derivatives may not have a direct influence on the response of the aircraft, particularly one as large as the Boeing 747, the pilot, in trying to attain a stabilized approach, finds the fluctuating airspeed a destabilizing factor requiring higher approach speed, larger throttle inputs, and larger glideslope deviations. The pilots remarked that such airspeed fluctuations are noticed in real-life encounters with turbulence, requiring all the above actions. Figure 5 shows the different airspeed variations encountered when flying through turbulence from models M1 and M3.

To evaluate the effect the simulator washout filters had on the pilots' perception of the turbulence, the simulator specific forces and angular rates were compared with those of the aircraft cockpit. It was found that although the magnitudes of the specific forces and angular rates were attenuated by the washout filters, the variation among the turbulence models was not altered appreciably. The largest differences came in the more restrictive simulator degrees of freedom, i.e. heave and pitch. In the former case, there was a significant subject effect in the standard deviation if the aircraft Z specific force (1.2 %) (as well as a significant treatment effect) but for the simulator motion the subject effect was not significant (16.8 %).

From these results several observations can be made:

- There did not seem to be any appreciable difference in the pilots' performance, or aircraft response between the formulations of models M3 and M4 (Equations 6 and 7).

- Without some modification the pitch gusts generated using Equation 9 were excessive and unrealistic. Any modification should take into account the wing downwash.

- Model M2 tended to produce the largest aircraft deviations at the early stages of the flight.

- The implementation of models M3 and M4 allowed realistic airspeed fluctuations to be simulated, and on an ILS approach were more realistic than the other models. It is possible that models M1 and M2 could be used to generate the gust time derivatives required to produce such airspeed fluctuations (Equation 14).

- The effect of the washout filters did not alter the performance results significantly.

Thunderstorm Downburst Wind Shear Evaluations

In studying the pilots' performance in flying through the thunderstorm wind shears, two aspects are examined. It is necessary to determine firstly if the ring vortex models produce similar pilot and aircraft responses, and secondly if there is any noticeable improvement in the pilots' performance over the course of the evaluations.

A similar analysis to that of the approach phase of the turbulence evaluations was carried out. In addition, several flight path deterioration parameters (FPDP) are calculated. These are similar to those in Reference [28]. By examining the variation of each parameter studied over the course of the six runs (seven including the no shear data) averaged over the six pilots, there was not found to be any significant change between the performance in the first run or the last. A similar result was found when the pilots' results were analysed separately. It is thought that the statistical population and the sample sizes are too small as to discover any learning trend. Also no instruction was offered during the tests although the pilots were encouraged to discuss the flights after each one.

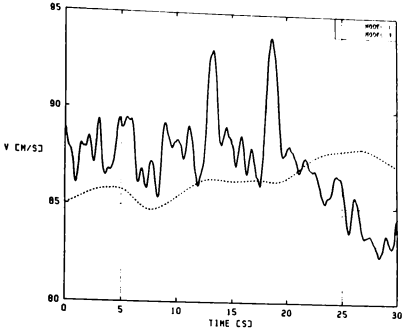


Figure 5: Airspeed Fluctuations for Models M1 and M3

The increasing headwind experienced by an aircraft approaching a thunderstorm downburst causes its airspeed to increase, and the consequent increase in lift leads to the aircraft rising above the glideslope. In an effort to attain a stablized approach, many pilots during their initial runs, in finding themselves above the glideslope for a given power setting, interpreted (through comments after each flight) the wind profile as an increasing tailwind. It was not until they had encountered the downdraught that they understood the nature of the wind field through which they were flying. Although presentation of groundspeed information to the pilot would immediately clarify such a situation, this observation emphasizes the insidious nature of the downburst wind shear. As stated above, there was no noticeable improvement in the pilot performance or aircraft response statistics over the course of the series of six approaches, despite this apparent learning process in understanding the wind field.

In addition, the capability of the ring vortex models to simulate the JAWS wind field was examined. It was found that the wind shears based on the JAWS $\overline{CD}$ scenario produced the most severe shears. FPDP's GS_+ and GS_- were reproduced well in both $\overline{AB}$ and $\overline{CD}$ cases, for all ring vortex models. During the course of the tests, it was observed that the pilots' primary task, on encountering the shear was to avoid undershooting the approach, and airspeed only becoming a primary consideration as it approached the stall speed. Consequently the primary indicators of the effect of the shear should be the glideslope and velocity FPDP's GS_+, GS_-, V_+ and V_-. Figure 6 shows these parameters averaged over all the pilots' results. It can be seen that all six wind shear models produce very similar GS_+ values, with very little spread in the SR2 and TR2 models. For the GS_- values, the SR2 model produces the greatest spread but _1 ($\overline{AB}$) and _2 ($\overline{CD}$) models are comparable within themselves. The V_+ parameter is produced equally well for the three _1 models, with some variation in the _2 models. Similarly for V_-, with TR2 producing the largest airspeed loss value.

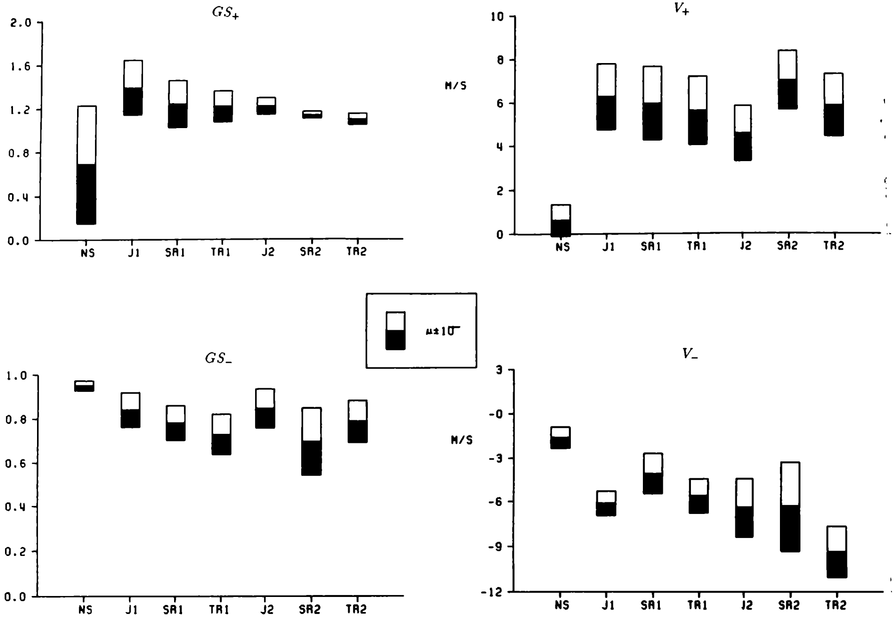


Figure 6: Flight Path Deterioration Parameters for Downburst Models

In comparing the single ring models with the triple ring models, there did not seem to be any significant difference in the statistics. Examination of the flight paths showed the glideslope and localizer deviations tended to occur more rapidly than the single ring models, in part due to the more intense, localised flow field produced.

Although the tests are not exhaustive, it may be understood from these results that the wind velocity field due to a thunderstorm downburst may be modelled using a ring vortex system, thus avoiding the use of larger, less flexible data sets. The wind shears used in these tests were all found by the pilots to be realistic. However the results presented pertain only to a very short portion of the flight phase. Exposing a pilot in a simulator to one single downburst cell may evoke a false sense of security, in that once a go-around, should it be required, has been initiated and the perceived threat cleared, the pilot will be safe. In reality however, several downburst cells can exist simultaneously in very close proximity to each other, and in clearing one cell, the pilot may inadvertently fly into another. A combination of cells would therefore be more desirable. The method of implementation of the ring vortex models used in this work is such that several cells may be included without very much increase in computation time.

Conclusions and Recommendations

It was found in the course of this work that the inclusion of spatial correlations in the turbulence models did not seem to have a significant effect on the pilot or aircraft. The inclusion of wind time derivatives in the equations of motion produces airspeed fluctuations which were shown to have a marked effect on the pilots' performance on the approach. The pitch gusts in models M1 and M2 were found to be excessive and this method (Equation 9) should not be used without modification. This simulation was applied to the largest aircraft in current airline service, the Boeing 747. The effects on a smaller, lighter aircraft may be somewhat different. This would be a suitable course of further study.

The downburst evaluations revealed the ring vortex models to be very capable in simulating the JAWS data. All models produced similar FPDP's, implying that such models may be used in the future as a useful training tool. Further research should be carried out in defining realistic boundaries for the various ring vortex model parameters; strength, radius, height, tilt, etc. Also several ring vortex systems should be combined to make a larger, more complex flow field. This can be done without an excessive computational overhead. Finally, further study should be made in implementing a turbulence structure on the ring vortex flow field. Although turbulence models have been included with such models in the past [21], they have been simply superimposed on the flow field. Previous work [29], indicates there to be a complicated turbulence structure associated with the downburst, and its associated gust front.

Acknowledgments

The author would like to thank the evaluation pilots who took part in this project. They were unpaid volunteers who displayed a high degree of interest and professionalism throughout the experiments. Without their expertise and dedication, this study could not have been completed. The author also wishes to thank the staff at the Canadian Airlines Pilots Association, Air Canada, and Boeing Canada for their help in recruiting the pilots.

References

[1] Houbolt, J. C., "Survey on Effect of Surface Winds on Aircraft Design and Operation and Recommendations for Needed Wind Research". NASA CR-2360, December 1973.

[2] "Airborne Low Altitude Wind Shear Equipment and Training requirements; Final Rule". U.S. Dept. of Transportation (FAA) 14 CFR parts 121 & 135, Docket no. 19110; Amdt. nos. 121-199, 135-127).

[3] "Low Altitude Wind Shear and its Hazard to Aviation". National Academy Press, Washington, D. C., USA, September 1983.

[4] Batchelor, G. K., *Homogeneous Turbulence*, Cambridge University Press, 1953.

[5] Etkin, B., *Dynamics of Atmospheric Flight*, J. Wiley & Sons Inc., 1972.

[6] Etkin, B., Personal Communication.

[7] Campbell, C. W., "A Spatial Model of Wind Shear and Turbulence for Flight Simulation", NASA TP-2313, May 1984.

[8] Etkin, B., "The Turbulent Wind and its Effect on Flight". AIAA Wright Brothers Lecture, UTIAS Review no. 44, August 1980.

[9] Van de Moesdijk, G. A. J., "Non-Gaussian Structure of the Simulated Turbulent Environment in Piloted Flight Simulation", Dept. Aerospace Engineering , Delft University of Technology, the Netherlands, Memo M-304, April 1978.

[10] Press, H., "Atmospheric Turbulence Environment with Special Reference to Continuous Turbulence". AGARD Report no. 115, April - May 1957.

[11] Reeves, P. M., "A Non-Gaussian Turbulence Simulation". USAF Technical Report AFFDL-TR-69-67, November 1969.

[12] Van de Moesdijk, G. A. J., "Specification of a Non-Gaussian Turbulence Model for On-Line Flight Simulation". Fokker Report no. A-161, May 1985.

[13] Dryden, H. L., " A Review of the Statistical Theory of Turbulence". Quarterly of Applied Mathematics 1943-45, January 1943.

[14] Tomlinson, B. N., "Developments in the Simulation of Atmospheric Turbulence". Royal Aeronautical Establishment TM FS-46, September 1975.

[15] Jewell, W. F., and Heffley, R. K., "A Study of the Key Features of the RAE Atmospheric Turbulence Model." NASA CR-152194, October 1978.

[16] Jones, J. G., "Review of Atmospheric Structures and Models. A Statistical Discrete Gust Theory Progress Note". Royal Aircraft Establishment TM FS-186, June 1978.

[17] _ Engineering Sciences Data Unit; Items 72026, 74030, 74031, and 75001.

[18] Zhu, S., and Etkin, B., "Fluid Dynamic Model of a Downburst". UTIAS Report no. 271, April 1983.

[19] Bray, R. S., "A Method for Three-Dimensional Modeling of Wind Shear Environments for Flight Simulator Applications". NASA TM-85969, July 1984.

[20] Etkin, B., Personal Communication.

[21] Woodfield, A. A., "Real-Time Simulation of Thunderstorm Wind Shear, Turbulence, and Steady Wind". Royal Aircraft Establishment, Bedford, U.K., 1984.

[22] Frost, W., H. P. Chang, K. L. Elmore, and McCarthy, J., "Microburst Wind Shear Models from Joint Airport Weather Studies". DOT/FAA/PM-85-18, June 1985.

[23] Reid, L. D. and Nahon, M. A., "Flight Simulation Motion-Base Drive Algorithms: Part 3 – Pilot Evaluations", University of Toronto, Canada, UTIAS Rept. 319, Dec. 1986.

[24] Hanke, C. R. and Nordwall, D. R., "The Simulation of a Jumbo Jet Transport Aircraft, Volume 2: Modeling Data", NASA CR-114494, Sept. 1970.

[25] Hanke, C. R., "The Simulation of a Large Jet Transport Aircraft", NASA CR-1756, March 1971.

[26] McDonnell, J. D., "An Application of Measurement Methods to Improve the Quantitative Nature of Pilot Rating Scales", *IEEE Transactions on Man-Machine Systems*, vol. MMS-10, no. 3, September 1969, pp. 81-92.

[27] Seaver, D. A., and Stillwell, W. G., "Procedures for Using Expert Judgment to Estimate Error Probabilities in Nuclear power Plant Operations", U.S. Nuclear Regulatory Commission, NUREG CR-2743, pp. A-1-A-12.

[28] Frost, W., Turkel, B. S., and McCarthy, J., "Simulation of Phugoid Excitation Due to Hazardous Wind Shear". AIAA-82-0215, January 1982.

[29] Lewellen, W. S., Teske, M. E., and Segur, H., "Turbulence Transport Model of Wind Shear in Thunderstorm Gust Fronts and Warm Fronts". NASA CR-3002, May 1978.

89-3285-CP

CIG DATA BASE DESIGN ISSUES

Patricia A. Widder and John L. Richard
McDonnell Douglas Helicopter Company
Mesa, AZ

ABSTRACT

Designing, generating, and critiquing visual and sensor data bases can be a very subjective process. This subjectivity is due to the fact that the definition of data base quality varies considerably from one person to another, and many people are involved in the design, generation, acceptance, and use of a data base. The design and synthesis of a quality data base are complex processes that entail the examination of several difficult and often conflicting issues. This paper will discuss these different issues and the impact they have upon the data base design and development process.

INTRODUCTION

Briefly stated, the problem addressed by this paper is the increasing cost of data base development to support computer image generators (CIG). One factor increasing the data base development cost is the data base rework that is often necessary before a data base is accepted. An improved data base design process should help to minimize the amount of necessary rework. Decreasing data base design time while simultaneously decreasing the amount of rework has the potential to significantly reduce the overall data base development cost.

The advances made in CIG and related data base generation system technology over the past 15 years need no enumeration here. However, the cost of designing and generating a data base which makes maximum use of the capabilities of a particular CIG has gone up, not down. One should not infer from the preceding statement that the fault lies mainly with the CIG manufacturer for not improving data base technology with the same fervor as the CIG technology. Although this is part of the problem, another major contributor is the increasing sophistication of the simulator user and the tasks that must be performed with the simulator. It would not be incorrect to state that as the number of features and capabilities of CIGs increase, the cost of data base generation for that CIG goes up as well, especially if that CIG's unique features are to be fully exploited. CIGs with many different features and capabilities also require the simultaneous resolution of many different constraints by the data base engineer.

WHAT IS A QUALITY DATA BASE?

It is probably safe to say that the determination of data base quality is a subjective process determined by many different people. A user may be a helicopter pilot or copilot/gunner, fighter pilot, bomber pilot, weapons officer, tank gunner or driver, naval officer, trainer, researcher, or engineer. Each user will have his own definition of quality and will be looking for, and judging certain aspects of, a data base that is unique to his own special needs. The increasing number of users, combined with their widely varying missions and multitude of sensors and visual enhancement devices, mandates the use of semi-custom visual and sensor data bases. In addition to the data base engineer's standards, the many user's requirements and expectations (and hence the user's perception of the quality), must be analyzed and incorporated into the finished data base.

The goal of the data base engineer is to efficiently design a data base to optimally support a set of specific tasks for a simulation. This includes taking full advantage of the CIG system's capabilities and working around its weak points. The process of optimizing a data base may cause the data base engineer to give up some of the 'real world' (perhaps even lacking in some aspects of realism) in order to provide more of the cues required for the particular task to be performed. This can often be the case when designing a data base for a specific training task.

SIMULATION CONSIDERATIONS

The translation of the user's expectations into a data base design can be broken down into two interrelated categories: the hardware simulation configuration used and the tasks to be performed with the simulator. A well designed data base is one which optimizes both components to meet all requirements. As one would expect, the specific CIG system used to support the simulation has a dramatic impact on the appearance of a data base.

MISSION RELATED CONSIDERATIONS

The first step of the data base process to determine who the user is and the purpose of the data base. The military services worldwide have a broad variety of aircraft, ground vehicles, and ships which are employed both independently and in conjunction with each other to perform many different types of missions. The device being simulated, and its specific role define what must be included in a data base. Once the simulator device, its role, and the kind of mission to be performed have been defined, the likely threats and targets can be determined. The design of correlated data bases can be considered the simultaneous design of multiple data bases, all of which will function as a single data base. It is the job of the data base engineer to provide sufficient cues to allow the task to be successfully completed.

The following are a few of the many questions which must be answered during the data base design phase if a high quality data base is to be produced:

What device is being simulated? A data base will be designed differently depending upon whether it is a fixed wing aircraft, ground vehicle, rotorcraft, or ship. Each of these device classes may be capable of supporting several different types of missions. For example, possible missions for a rotorcraft might include scout, attack, supply, search and rescue, and covert operations. For each of these missions many of the requirements are the same, but, each type of mission has its own unique requirements as well. Optimum performance, from both the simulation device and the training imparted, can best be achieved from data bases which are designed for that specific type of mission.

What are the speed and altitude ranges for this simulation? Another important consideration in defining the required cues for a visual or sensor data base is specifying the altitudes and speeds at which the missions will be flown. The many different types of military aircraft, armored vehicles, and ships all have several unique needs that dictate custom data bases for highly effective training, mission rehearsal, and person-in-the-loop engineering design.

Low speed and NOE flight necessitates many features that are of grave importance to a Cobra pilot but have little impact on a KC-10 pilot operating at higher altitudes. Highly detailed radar return attributes are a necessity to the F-111 or B-1B crew but are comparatively insignificant to the pilot of a CH-46 cargo helicopter. A data base should be designed to support flight at a specified range of speed and altitudes. In addition, each gaming area within the data base should also be individually designed to support the speed and altitude range for that limited area. If a single data base must support a wide range of speeds and altitudes over the same area, it must be recognized that some serious trade-offs must be made.

What sensors (including OTW) will be required? Applicable avionics equipment and state-of-the-art sensors must also be simulated accurately. The increasing number of modes designed into digital moving maps, radar, NVGs, infrared, and other electro-optical sensors, inundate the data base engineer with a multitude of new information that must be added to correlate with what was once only out-the-window data.
Contour lines, in selectable increments, and a wide variety of symbols must be available for moving map displays. Many different radar return attributes and aircraft radar cross sections must be known in order to simulate Synthetic Aperture Radar, Real Beam Ground Mapping, or other radar modes. The modeling of accurate infrared signatures is necessary for realistic training, engineering design, and mission rehearsal purposes.

What is the scenario to be performed? This requires that one or more missions be specifically defined. Will the pilot need to perform any detection, recognition, and identification tasks? What are the possible encounters?

In the detection, recognition, identification arena, the accuracy to which friend or foe have been modeled for simulation exercises is very important. An AH-64 pilot or copilot/gunner must be able to distinguish the differences among T-80, M-1, Leopard 2, and Challenger tanks. Likewise, an F-14 pilot and weapons officer must be able to make the distinction between an F/A-18 and MiG-29. In many cases, targets and threats and widely varying terrain and atmospheric conditions can be found throughout the world. The cues necessary for reliable detection, recognition, and identification must be simulated accurately or an unfortunate degradation in training may result.

The order of battle also needs to be defined. This includes target and threat locations, friendly ground and air support locations, and target and threat definitions (e.g., type, level of interaction, weapon environment). In addition, the following characteristics need to be defined for both friendly and foe forces: flight/movement, configuration parameters, weapon characteristics, quantity, vulnerable point(s), and required accuracy.

What skills are to be practiced in the simulator? This also has a big impact on data base design. Data bases of the same geographical area may look radically different if one is to support the training of basic flight skills and the other is to support mission rehearsal. Examples of skills which may be taught or practiced in the simulator include basic flight skills, weapons management, route selection, situation assessment, mission planning, military operations, and mission rehearsal. An important part of data base design requires the data base engineer to determine the relative importance of the features to use in case of CIG overload.

What geographic area is to be used in this simulation? The geographic area needs to be defined for the entire data base as well as for all navigational and gaming areas. The navigational areas are those areas which must support ingress and egress to/from the target area. There should be gaming areas designed into the overall data base to support target acquisition and target attack areas. Data base features should be selected to support the specific tasks to be simulated in the different gaming areas.

If important to the simulation task, specific attributes of the geographic area should be determined during the data base design. Such attributes include vegetation (primary and secondary), terrain type, and natural and cultural features. All landmarks (navigational and otherwise) should be identified. Aircraft control data (e.g., refueling points, flight corridors, etc.), also need to be determined prior to data base design.

What are the environmental conditions? Environmental conditions include visibility, weather (e.g., fog, rain, snow, scud, clear, cloudy, and temperature), time of the year, time of day. Will the weather be changing within a single simulation task? The actual clock hours to be covered during the simulation task should also be defined if important. Although some of these may be parameters passed to the CIG during initialization, it is important that the data base be designed and developed with these conditions in mind.

Are correlated data bases a requirement? If correlated data bases are required, the minimum degree of correlation and interoperability must be specified before data base design can begin. Additional details on correlated data bases will be presented later.

SIMULATION SYSTEM RELATED CONSIDERATIONS

IMAGE GENERATOR. It is obvious CIG functions such as priority determination are dependent on the CIG architecture, and must be considered during the data base design phase. It is not as commonly understood that one Brand X CIG can have a vastly different configuration than another. The same model CIG may have different polygon capacities, different channel configurations, different dynamic coordinate set configurations, different on-line memory capacities, and different mission functions being performed in the CIG's front-end processor. When all this is taken into account, one can understand how data bases for two different Brand X CIGs may differ more than between a Brand X and a Brand Y. These parameters must be used to define the bounds of the final data base. Initial data base density calculations cannot be made until the parameters are specified. The following paragraphs review some of the parameters which should be identified, understood, and defined before beginning the data base design process.

CIG Architecture. Everyone is aware that the basic architecture of the CIG has a dramatic impact on the data base design. Three CIG architectures which are currently available are the conventional priority based system, Z-buffer system, and photo-based system. Each type of system has its own advantages and disadvantages. For example, a data base designed for use on a Z-buffer system will probably not work at all on a priority based system. Certainly it will require significant modification.

Some CIGs support points and point lights; others don't. Some CIGs can display lines; others can't. Objects on some CIGs must be closed and convex; it is not a requirement for others. The capacity of the CIG's on-line memory also plays a significant role in the design of a data base since it can only display data that it can access in its on-line memory. The rate at which this memory can be updated is also important.

Polygon Capacity. The polygon capacity is usually one of the first questions asked about a CIG. No matter how many polygons a system can support, it is never enough. The data base engineer must decide how to allocate those limited polygons between the terrain, culture, fixed models (both cultural and tactical), and dynamic coordinate sets. Are there any rules about how the polygons are distributed - must they be evenly distributed across all channels or can they be concentrated in one or a very few channels? Will the system support variable density terrain? Even if it does, how does its use impact the data base design?

Load Management. While a load management feature performs a valuable function in trying to minimize distracting overload anomalies, it is not the panacea that many perceive it to be. Load management cannot adequately perform its intended task if the data base does not have the proper design. Generally speaking, the numbers and complexities of moving models, universal features, special effects, fixed models, and amount of terrain and cultural detail can seriously impact the performance of load management. Level-of-detail (LOD) switching, blending, and judicious moving model implementation must be utilized in order to minimize the degrading effect of load management.

Mission Functions. Mission functions may be performed in either the special purpose or general purpose portion of the CIG. Typical mission functions include: line-of-sight, height above terrain, collision detection, threat occulting, and laser range finding. The data base engineer must understand the impact these functions will have in terms of the CIG processing time. If the data base is not designed, developed, and tested with these functions in mind, the additional processing time may be enough to cause the CIG to overrun the frame. Environmental effects such as fog, haze, visibility range, clouds, and horizon glow may also have an impact on the CIG processing time, and so, should be considered during the data base design.

Sensor Characteristics. Does the CIG support sensor simulation?
If so, what sensor types and characteristics does it provide? Does a separate data base need to be developed to support sensors or can both characteristics be combined in a single data base? What needs to be provided as part of the data base to support accurate sensor simulation (e.g., specific signatures)? Are additional viewpoints required to support this sensor simulation?

Dynamic Coordinate Sets. What are the limitations? How many unique and simultaneous dynamic coordinate sets does the CIG support at any time or for an entire mission? Does the system support articulated parts, and if so, can they be nested? Is there any advantage to specifying some coordinate sets as only having three degrees or freedom instead of six degrees of freedom? Are there any priority limitations unique to dynamic coordinate set processing which must be considered?

Special Effects. On some systems,special effects are considered to be a subset of dynamic coordinate sets. One of the difficulties dealing with special effects is the inconsistency of their implementation on different CIG systems. A special effect on one system may be treated as an ordinary data base feature on another. In addition, there may be different types of special effects (e.g., color change, and texture movement) which must be considered.

Texture. A photographic type of texture is one of the newer features supported by many CIGs. There are a number of questions concerning texture which need to be answered during the data base design process. Is the texture color or monochrome. Full color texture patterns can introduce a new set of problems to be solved. How many on-line texture maps are supported? What is the resolution of the texture maps? Can the maps be divided into submaps? If so, what are the limits of their resolution? Do the texture maps have multiple LODs? If so, what are the rules concerning their use? What are the rules guiding the placement of texture maps on the various data base components? Does the system support dynamic texture map update?

Channel Configuration and Visibility Range. The specific channel configuration and visibility requirements to be used for a simulation task also have a dramatic effect on the constraints to be satisfied during the data base design task. In addition to the constraints imposed by the CIG, the channel configuration also defines the supported instantaneous field-of-view (FOV). The FOV required to support the simulation task is one of the basic inputs which impacts the data base design. The data base will be designed differently depending upon the FOV and the visibility ranges. This means that even if two CIGs are equivalent, but the FOV is different, the data bases may still need to be designed differently.

It is important to know in advance whether any of the sensors have a variable FOV. It is not an easy task to design and develop a data base to support an FOV ranging from 0.45 to 40 degrees and a visibility range up to 20 nm. It is almost impossible to take an existing data base and modify it to support a very different FOV without it overloading or making it almost useless for the original tasks to be performed.

DISPLAY SYSTEM. Another area which must be considered is the display system being used. In this case, the term display system includes both the projection system and display environment. It is often assumed that if the same data base and CIG are used, the resulting image will be the same for any type of display system. This is not true.

If the output from the CIG is to be projected onto the surface of a dome, it may be necessary to correct for the shape of the dome, distortions of the optical system, and the off-axis position of the projector. Some CIGs provide the capability to compensate for all of these factors through predistortion of the image. The solution used by some systems is to continually subdivide the polygons until projection of the edges on the dome surface appears linear. As can be seen, using this process can result in a proliferation of polygons.

The data base engineer must also know the resolution provided by the display system. There is no point in designing a data base for which the CIG must output polygons that are too small to be seen due to the limited display resolution. In addition, the limitations of the projection system also need to be considered. For example, the dynamic range of the light valve must be kept in mind when assigning and tuning colors.

CORRELATED DATA BASES

The correlation and interoperability of data bases is rapidly becoming one of the critical components needed to accurately simulate mission related tasks, especially those involving more than one person. The increasing use of sensor systems is pushing simulator requirements toward multiple environments for each crewmember. These subsystems support out-the-window (OTW) displays, infrared (IR) displays, radar displays, moving map displays, non real-time operator displays, and real-time software including specialized functions such as line-of-sight (LOS), moving model control, and scoring.

In addition to multiple environments for each crewmember, the ability to perform a full mission simulation and evaluation is becoming increasingly important in the training of pilots. Full mission simulation in this case not only includes multiple cockpits, but also instructor consoles, tactical situation displays, mission planning and the ability to review and critique each training exercise upon its completion. This full mission simulation demands much more correlation than simply matching the geometry of models seen on visual and IR displays.

The networking of simulators both within a single simulation facility and between different facilities, increases the problems brought about during the correlation of these multiple environments. Just as each simulator must correlate all subsystems, a simulation facility (or networked facilities) must provide the capability of interoperability, that is, to correlate the data bases for the subsystems simultaneously. This can be 'fairly easy' as long as the same hardware and software is used on each simulator. However, the problem gets much more complicated when different hardware and software components are used.

Data base correlation can mean different things depending on the specific situation. It is very difficult to specify the degree of correlation required for each data base feature or class of features in any particular simulation task. This is due to many factors which need to be considered while determining the type and degree of correlation. For example, the altitude an aircraft flies will determine how accurately features on the ground need to be represented. A high altitude bomber pilot does not care if every tree is in the same location; he may only care that a forest can be seen. On the other hand, for a helicopter (using both OTW and IR) flying NOE, the trees need to be in the same location on both displays.

Real World Correlation. Does the data base resemble the real world or the source from which it was generated? What if the source data is not correct? Do the visual cues needed to fly the mission exist in the data base (e.g., trees that provide height and motion cues)?

IR Correlation. Do the IR features provide the pilot with correct cues showing the effects of surface material, temperature, sun angle and other IR characteristics? Is a high fidelity IR model a requirement (which may bear little resemblance to the visual model) or is it sufficient to color a visual model different shades of gray?

Radar Correlation. What degree of correlation is needed between the radar display and the OTW scene? If a high degree of correlation is required, is the user willing to settle for polygonal accuracy in the radar display?

RESULTS OF THE DATA BASE DESIGN PROCESS

The data base engineer uses the answers to the above (and many more) questions and previous experience to design the initial data base. There are many different specialized areas which may need to be considered when designing an optimal data base for any particular set of circumstances. Data base engineers with different backgrounds (different levels of expertise in the various related specialized skills) will design and generate 'the same' data bases with widely varying results. This does not mean to imply that all data base engineers should have each of these skills. However, as more of these skills reside within a data base engineer or a data base modeling team, the 'better' the resulting data base will be. This is due simply to the fact that the data base engineer has a better understanding of the problem (both mission and CIG related) and can evaluate the various alternatives in terms of their impact on the data base as a whole. These skills are not necessary for every data base built. For example, a private pilot may have a better understanding of the cues necessary to support a flight trainer but would not have any advantage developing a data base for a ship simulator.

The data base design process should result in a set of specifications for a series of data base parameters. These parameters can be divided into five categories: terrain, surface (2D) features, 3D cultural features, 3D fixed tactical features, and dynamic coordinate sets (moving models). Texture also needs to be considered during the data base design process. Although it is an important design factor, it is often overlooked until later in the data base development cycle. The following paragraphs describe some of these parameters.

Terrain. Perhaps the first criteria which must be determined is how closely the terrain must match the real world or the source data (goodness-of-fit). Next the minimum and maximum terrain polygon densities need to be defined. Although the number of terrain polygons are usually minimized, there are circumstances when additional terrain polygons are required. The terrain density should not necessarily be the same for the entire data base. The degree of regularity of terrain polygons should also be determined. Any restrictions on terrain polygons also need to be specified. The number of terrain LODs, their visibility ranges and the transition ranges between LODs should also be defined. In addition, if terrain polygons are entirely covered by a single culture polygon, it may be desirable to replace the terrain polygon by the culture polygon.

Surface Features. As is the case for terrain, the number of LODs supporting surface features also needs to be specified. For each LOD, the visibility range and transition ranges between LODs also need to be defined. The desired accuracy of the various types of surface features also needs to be specified. Any restriction on surface feature polygons also need to be defined. The maximum number of surface feature polygons for a specified area also needs to be defined. In addition, the relative importance of different types of surface features in different areas of the data base need to be determined. Again, these will probably vary within the same data base.

3D Feature Representation. Although this can encompass fixed cultural and tactical 3D features as well as moving models, the specific answers will probably be different. The correct selection, definition, and placement of 3D features is one of the most important aspects of data base design.

Feature selection includes not only the correct model but also the appropriate LOD(s) for that model. Note, however, that the most important types of models for any given area will probably differ throughout the data base. Feature definition is basically the geometry of the model and its attributes on a submodel basis (object, polygon, vertex). The number of polygons for each LOD of each model type as well as the visibility ranges and transition ranges between LODs also need to be specified. The required accuracy of each model will be an important factor in determining the polygon count. Since feature placement includes determining the optimum location for each model, the desired densities of the different model types in the different areas of the data base also needs to be specified.

DATA BASE REQUIREMENTS

A clear and agreed upon understanding of the user's requirements and expectations should be performed at the beginning of the data base design process. In an ideal world the answers to the above questions would be provided in the data base requirements specification. If this does not happen, it is up to the data base engineer to make all assumptions clear to the user. It is vital to get the user involved as early as possible to avoid any incorrect assumptions being made by the data base engineer or the users. The initial meetings between the data base engineer and the users should lead to a sound and comprehensive data base requirements specification. A lot of misunderstandings are rooted in ambiguous or nonexistent data base requirements. A requirements document should be written which has one, and only one, interpretation to everyone involved in the data base development and acceptance process as well as the data base users.

An experienced data base engineer has, and maintains, a thorough knowledge of available hardware and software capabilities to enable the full exploitation of a CIG's potential. When this experience is coupled with a logical data base design process, the chances are maximized that the CIG will be utilized to its fullest extent possible and the data base will meet both the user's expectations and requirements. It is also important to remember that there is little good in specifying an ideal data base which cannot be successfully implemented on the CIG system to be used. Image generator technology can be pushed only so far.

CONCLUSION

The data base engineer will usually be the person responsible for solving the bulk of the data base design problem for any simulation task. However, before this can be done most details such as the purpose of the mission to be performed, any important task parameters, and the hardware to support all the simulation subsystems must be defined. It falls upon the shoulders of the data base engineer to explain the trade-offs involved to all users. Ideally, before any data base design is begun, all parties involved with the simulation will sit down, examine the various trade-offs, and reach an agreement in terms of an acceptable degree of correlation for each of the subsystems involved. Failing to discuss these correlation issues will only result in disappointments or even an unacceptable simulation.

89-3286-CP

NIGHT VISION GOGGLES: TERRAIN & FEATURE SHADOWING FOR CIG DATABASES

Robert Geer
Software Engineer, CT Exploitation
Evans & Sutherland Computer Corporation
Salt Lake City, Utah

Abstract

Night vision goggles (NVGs) are the principal sensors used by the military to aid night flight. Experience has shown that NVG flight regimes are dangerous but that the risks can be managed with proper training and preparation. The military is looking to flight simulation to provide training in a minimal risk environment for transitioning aircrews to NVG flight.

NVGs amplify available light, and very dark areas remain very dark and visually impenetrable to the NVG wearer. Shadows are a primary source of danger to NVG aviation and must be represented accurately in the simulator. Since computer image generator (CIG) hardware does not dynamically compensate database brightness within an area where a shadow should be cast, an NVG-compatible CIG visual database must have accurately modeled shading and shadows of terrain relief on terrain, terrain on vegetation/features, and vegetation/features on terrain. This paper describes the techniques required to achieve the desired database appearance.

Introduction

The ability to survive in combat operations depends on both minimizing the opponent's capabilities and maximizing one's own. The very low level of ambient light available at night provides an excellent opportunity to escape detection, but the potential effectiveness of one's own activities is likewise limited. Artificial light, either visible or infrared, probably enhances the chances of detection more than it enhances the potential effectiveness of any operation that is undertaken. Night vision goggles amplify ambient night light levels without increasing the chance of detection and weapons engagement by the enemy. This preserves the chance of surprise, and increases the effectiveness of night operations. To fly low level profiles using night imaging devices which operate in the optical radiation portion of the electromagnetic spectrum, requires a thorough understanding of the night environment. This includes the relationships between ambient illumination, lumination*, the terrain, the night imaging device, and the human eye.

With night vision goggles (NVGs) now added to the list of sensors used by the military to aid night flight, it is becoming apparent that NVG flight regimes are dangerous, but that the risks can be managed with proper training and preparation. The publication of the Marine Corps Night Vision Goggle Manual(1) and establishing the Night Imaging and Threat Evaluation (NITE) Lab at the MCAS Yuma are significant steps in that direction. As NVG training evolves, the military will look towards flight simulation to provide transitioning aircrews with training in a minimal risk environment, thus increasing the chances of safe NVG flight.

The moon reflects 7% of the sunlight hitting it, and such variables as the altitude of the moon above the horizon, the phase of the moon, atmospheric conditions, and the proximity of artificial light further influence the limited amount of light that eventually becomes usable ambient light in a given area of operation. NVGs amplify available light, but very dark areas will remain very dark and visually impenetrable to the NVG wearer. Shadows, which limit even more of the limited available light, are a primary source of danger to NVG nap of earth (NOE) aviation and must be represented accurately in the simulator. Computer image generator (CIG) hardware which is currently available does not dynamically compensate database lumination in areas where shadows should be cast. To be effective, NVG-compatible IG visual databases must accurately model shading and shadows of terrain relief on terrain, terrain on vegetation/features, and vegetation/features on terrain.

This paper demonstrates the techniques which give a database the desired appearance necessary for NVG simulation.

Moon Shading

CIGs have the capability of shading terrain, vegetation, and feature constructs based on the angle towards or away from the sun (called "the polygon's sun angle") of each polygon of each construct. In fact, it is typical for at least two sun illumination curves to be available, one for clear, sharp contrast days and another for hazy or overcast, low contrast days. However, moon shading requires several illumination curves, running the gamut from no contrast to very high contrast to account for the subtle influences from weather and atmospheric phenomena.

The moon phases are the first obvious influence on the amount of moonlight that contributes to available light at night. Other contributions to ambient light are made by the stars and the airglow* of the night sky. Clouds, precipitation, water vapor due to high humidity, and particulate matter such as blowing sand or dirt, are all obscurant, and reduce the amount of available light, thus influencing the contrast between light and dark

* *Illumination* is the amount of light incident upon a surface. *Lumination* is the amount of light reflected from the surface.

* The term *airglow* refers to the observable light that originates in the high atmosphere and is the result of photochemical reactions of gasses.

areas. Some of these conditions may diffuse the light so it appears less directional.

The presence of snow will also drastically change the apparent available light because it is so much more reflective than earth, foliage, or paving. It is difficult, if not impossible, to predict the contrast for all combinations of conditions. Further complicating the prediction of contrast is the presence of artificial light in urban and suburban areas and other specially lit areas such as around isolated factories, power plants, mines, and highway intersections or interchanges.

Rather than try to predict the various contrast requirements for each training scenario, it is more flexible to provide a family of contrast curves which may be changed "on the fly" during the course of the scenario. Such a family of curves is illustrated by the set of generic functions shown in Figure 1.

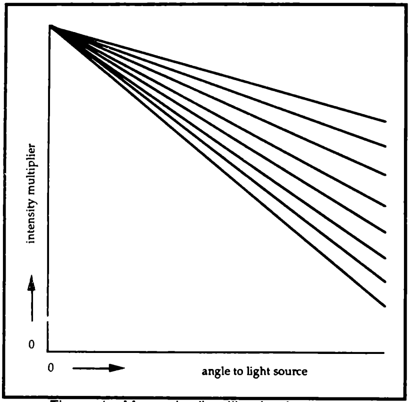


Figure 1. Moon shading illumination curves

The bottom function in Figure 1 represents a very clear night with very bright moonshine creating a high degree of contrast between surfaces directly lit by the moon and surfaces pointing away from the moon. The top function, on the other hand, represents a night with very little moonlight. In this situation, most of the available light is starlight and airglow, which is very diffuse, resulting in very little difference in contrast from one side of an object to another. Although it is not shown in Figure 1, a totally flat curve is available which may be used for the new moon phase when no moonlight is available and natural ambient light is the most diffuse.

It should be noted that there is more than one factor controlling CIG scene brightness. Sun angle illumination is determined by algorithms using curves such as those in Figure 1. Overall scene illumination is determined from a user input intensity multiplier. To simulate a particularly dark night, the overall intensity factor might be set to .5; the combined effect of choosing .5 and a curve whose range is 1.0 to about 0.6 is an intensity range of 0.5 to about 0.3.

It is important to have the capability of switching "instantaneously" between the the various curves. Sudden increases in brightness and contrast simulate the effects of lightning. Similarly, relatively short duration decreases in brightness and contrast simulate small clouds momentarily obscuring the moonlight.

Terrain Shadowing

Shadows cast by terrain onto terrain, such as those formed when the moon is positioned low on the horizon, causing a high ridge to shadow an adjacent valley floor, ridge line, or hill, are a cause of much of the danger in NVG flight. A typical example is a low hill completely shadowed by a larger hill. Even with NVGs, there is not enough ambient light to differentiate the existence of the low hill, and thus be able to consistently avoid flying into it. The NVG-compatible database must adequately represent terrain shadowing to be effective.

As noted in the introduction, current CIG design does not provide for dynamically cast shadow determination. However, current CIG design does provide for geo-specific* texture through expanded online texture memory and the real-time paging of texture maps from offline storage into the online memory. This is the perfect vehicle for modeling shadows cast by terrain onto terrain, since terrain shadows are indeed geo-specific.

A terrain shadow mask is produced by calculating the terrain shadows from terrain profiles and a given moon elevation above the horizon. The process results in a set of shadow masks which indicate the shadowed (dark) and unshadowed (light) areas of the database. These masks are then processed into texture maps for the terrain levels of detail modeled in the database (see Figure 2).

The Evans & Sutherland ESIG-1000 uses an improvement of the same texture design proven in its predecessor, the CT6, for the last four years. This design provides the capability for each polygon in the database to be textured with a unique combination of any four individual texture maps from online texture storage. Each of the four maps may be sized and contrast scaled separately, and then blended onto the polygon in real time. Geo-specific texture and shadow texture can be macro-scaled onto the polygon using two of the four available maps, which leaves two more maps which may be scaled in the medium (bushes and trees) and micro (grass and gravel) scaling ranges. Changing the shadow maps for a different moon position does not impact the other texture maps at all; reload the shadow maps and a whole new scenario is

* The term *geo-specific* refers to the use of terrain-specific, map-correlatable texture maps that have high fidelity to real-world terrain. See references (2) and (3).

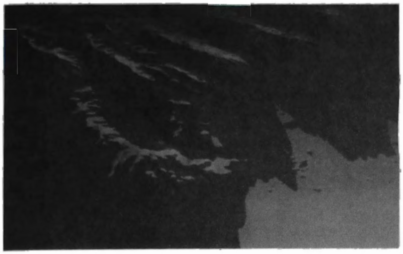


Figure 2. This view of a moonrise is looking slightly south of true east. The mountain ridges and peaks cast terrain shadows on adjacent ridges and the valley floor. The photograph was prepared from an ESIG-1000 image generator running in real time.

immediately available. Because the texture maps are all independently scaled before being applied to the polygon, the shadow map may be scaled much larger than the geo-specific texture, requiring many fewer maps for a change in moon position than if the shadow information was integrated into the geo-specific texture. Consequently, both online and offline storage requirements are reduced, shadow map calculation time is reduced, and map load time is much shorter.

If specific database requirements dictate, certainly the shadow mask may be combined algorithmically with the geo-specific texture. This leaves three independently-sized and contrast scaled texture maps available for whatever medium and micro-scaled texture is required. Scenario convenience is traded for scene content flexibility.

It should be emphasized here that the texture map scaling mentioned above applies to both size and contrast. Just as the texture map sizes are independently scaled to properly represent the shadows cast by in the real world, the contrast of the four maps is also independently controlled. A shadow doesn't obscure the area it covers; when the shadowed area is entered (or flown over), there must still be detail in the darkness which the NVG-wearer can pick out as the automatic gain of the NVG device improves the apparent contrast of what is viewed. The contrast control designed into the four texture map implementation provides the flexibility to achieve a wide range of terrain shadow obscuration without having to recalculate the shadow maps.

The important thing is that the use of a shadow map in no way limits the extremely wide range of the texture capability of the CIG, but it makes a tremendous contribution to potential training effectiveness.

Feature Shadows

Once the proper shadowing for the terrain is achieved our attention turns to vegetation on the terrain. Tree shadows can have important negative influences on flight with NVGs. The overwhelming majority of the trees in a CIG database are actually instantiations of a standard tree or a tree selected from a basis set* of trees (see reference (4)) This is an important mechanism for data compression in the visual database. Modeling tree shadows to fit every terrain slope and aspect has always increased basis set size and reduced the storage savings of the basis set concept. However, the introduction of photo-repeating texture (see reference (2)) into the database designer's toolkit suggests the elimination of this compromise. We can now realize of the full savings potential of the use of basis sets of trees by using "free" shadows provided by photo-repeating texture.

Photo-repeating texture is texture which is derived from a photographic source and processed so that its left and right edges match and its top & bottom edges match; if copies are laid edge-to-edge the seams should not be evident. The application of photo-repeating texture prepared from a photo of sagebrush and piñon results in a collection of spots on the terrain polygon which provide excellent small to medium scale flight cues. These spots are irregular in size and location, with a random appearance that only nature can achieve. However, the scale of these spots is not readily apparent to the pilot, which depreciates their value as altitude cues.

Since this texture repeats across the terrain, the trees may be instanced onto the terrain basis set polygons at locations where the spots occur, using the appropriate size of tree to match the size of the spot. A dual advantage is realized in that the trees are provided with "free" slope and aspect correct shadows, and the scale of the spots is evident from the size of the trees. Typically, the number of trees budgeted for each basis set polygon is less than the total number of texture spots, so the trees are distributed over each polygon by placing each tree on a spot that is randomly selected from the list of all of the spot locations on the texture map. The appearance of each basis set polygon is well differentiated from the others, thereby avoiding an "orchard" appearance of tree placement.

The distribution of trees on the terrain basis set appears more natural and each tree has a shadow that is properly placed on the terrain. The impact on the database budget is nil, since the same number of trees would be placed anyway. The cost is the one-time, offline cost of determining the acceptable locations for

* The term *basis set* refers to a carefully designed set of database components that are used frequently in the database. In particular, the terrain polygons are members of the terrain basis set. In a large area database the use of basis sets as opposed to the much more online and offline storage intensive compiled-in-place terrain polygons becomes an essential consideration in database design.
See reference (4).

tree placement in a scale compatible with the texture map's application to the terrain.

The photo-repeating texture map can be processed with a paint-type drawing software program to effectively "stretch" the texture spots in a specific direction to simulate low moon position shadowing. Since the four texture maps (one of which is the terrain shadow map) are applied to the polygon individually, the tailoring of the photo-repeating texture map for different moon positions is simply a matter of reprocessing a single texture map.

Buildings and other cultural features which cast shadows present a different sort of challenge when designing an NVG-compatible database. While trees are placed over wide areas of the database, with several occurring on any given basis set element, cultural features must be placed carefully and uniquely, and photo-repeating texture cannot be used to provide shadow detail. There are several strategies, however, for modeling shadows.

If geo-specific texture is used on the database, the feature can easily be placed so that underlying texture can provide the shadow. If no shadow is present on the texture, one can be "sketched" in using paint-type software.

If one does not wish to modify or use the texture for shadows, the feature may include a single shadow or set of shadows modeled with polygons. If the feature's shadow must appear correct for several significantly different moon positions, a new ESIG-1000 feature known as static model systems (SMS) may be exploited. This new feature provides models containing several different shadow configurations, similar to a high resolution dynamic model of a movable-wing F-14 that is modeled with several wing positions. Only one wing position is used at a time; similarly, only one specific user-chosen shadow of the SMS model will be processed and displayed during the simulation scenario. While the SMS model is similar to the dynamic model in concept, its processing has been optimized to eliminate the high front-end processing time required of models designed specifically for use on dynamic coordinate systems.

Terrain Shadowing of Features

Once the shadow masks are produced from the terrain elevation data, they may be used to determine whether a particular terrain polygon is shadowed or illuminated. Although the polygon itself is shadowed by the shadow map, such features as trees, road and river segments, cultural features, etc., that lie on the polygon must still luminate properly. For those terrain polygons which are completely shadowed or illuminated, the appropriate dark or light terrain basis set polygon with its instanced trees and features can be offset into place. For those terrain polygons which have both shadowed and unshadowed areas, the shadow mask also indicates how each feature should be shadowed or illuminated to be consistent.

The SMS concept can be exploited to provide scenario-specific shadowing of features. For a specific set of moon positions, the SMSs can be used to choose between shadowed or illuminated terrain basis set polygons that will be either completely shadowed, or completely illuminated. For those basis set polygons which straddle a shadow edge, SMSs can be used to select between shadowed and illuminated versions of individual trees. The same approach can be extended to cultural features. Alternately, selected features may be compiled in place rather than instanced to provide a properly shadowed or luminated feature.

Summary

The use of NVGs in simulation has presented new and unanticipated challenges to the database designer. CIGs whose design includes texture paging readily provide excellent simulation of terrain shadowing through straightforward extension of the geo-specific texture capabilities. The generality and flexibility of applying four different texture maps per polygon at any desired combination of scales means the extension *supplements* the available tools already in the texture toolkit. There is absolutely *no need* to suffer the elimination of any part of such commonly used texture assets as a combination micro, medium, or macro scaled texture to add terrain shadowing to a database.

A similarly innovative application of the general capabilities of the CIG also offers good to excellent solutions to the challenge of satisfying the NVG-specific requirements of lumination contrast control and feature shadows. Multiple lumination shading curves offer the contrast control needed to satisfy any scenario specific requirement. Multiple solutions to feature shadows gives the modeler the means to satisfy most database content budgets and still present the proper visualization to challenge NVG trainees.

This paper has shown that currently available CIG hardware is capable of providing excellent simulation of NVG flight conditions, and ongoing research and exploitation will no doubt continue to improve upon these capabilities. The simulation industry is dedicated to providing military flight simulation programs with the tools required to maximize the pilot's preparedness for all flight conditions.

References

1. "Night Vision Goggle Manual", United States Marine Corps, Marine Aviation Weapons and Tactics Squadron One, MCAS, Yuma, Arizona 85369-6073

2. Clark, L. Charles and Thomas C. Brown. "Photographic Texture and CIG: Modeling Strategies for Production Data Bases", *Proceedings from the 9th I/ITSC*, November 1987.

3. Wilkerson, Howard. "Scene Realism: The Synergy of Data Base Technology and CIG Hardware", *Proceedings from the 10th I/ITSC*, November, 1988.

4. Costenbader, Jan. "CIG Data Bases in an Instance: Bits & Pieces", *Proceedings from the Image III Conference*, May 30 - June 1, 1984.

Acknowledgments

The author would like to thank the following Evans & Sutherland personnel for their help and encouragement in producing this paper: Howard Wilkerson, Thomas C. Brown, and Chuck Clark of the CT Exploitation group; Mark Green and Rulon Nye of the Display Systems group.

About the Author

Robert W. Geer is a Software Engineer for Evans & Sutherland's Simulation Division where he is currently assigned to the CT Exploitation Group. He received his BS degree in Mathematics from Idaho State University. His exploitation initiatives include investigations into infrared sensor simulation, infrared sun shading, basis set juxtaposition visualization improvements, and mission rehearsal requirements. Previously, he was a project engineer building and delivering Evans & Sutherland image generators, and before that wrote data reduction and diagnostic software for hospital respiratory care departments.

89-3288-CP

STEREOPSIS AS A VISUAL CUE IN FLIGHT SIMULATION

Reed P. Tidwell
Evans & Sutherland
Salt Lake City, Utah

Abstract

This paper discusses stereopsis, and its role in visual systems for flight simulators. The role of stereopsis as a depth cue, and ways that stereo images may be created and presented are outlined. The limits of stereopsis as a visual cue in the simulated environment are quantified by experiments performed with an Evans & Sutherland ESIG-1000 computer image generator (CIG). The focus of the experiments is to isolate sensitivity to stereoscopic cues. Costs and potential benefits of stereopsis are discussed. Results indicate that stereoscopic cues created by a simulator are valid to distances which would make them valuable for several applications, and that further study with stereoscopic simulators is justified and necessary.

Background

Stereopsis is the mind's transformation of two slightly different images into a single view of the world with stereoscopic or three-dimensional depth. Sir Charles Wheatstone was the first to create stereoscopic images from planar images in 1833. He did this with the mirror stereoscope into which one could look and view hand drawn stereo pairs. The differences in the sketches viewed by left and right eyes created the sensation of depth. Wheatstone published his discovery in 1838. The following year photography was invented and soon, stereo photographs were made. The same concept was later applied to motion pictures and is now being applied to computer graphics, among other things. Today stereoscopic flight simulators of very high quality can be, and are being produced.

The Role of Stereoscopic Cues

The stereoscopic visual cues, convergence and disparity are two of many visual cues to depth. The combination of these two physiological cues constitutes what is referred to in this paper as stereopsis.

In order for an object to be seen as a single image by the brain, the central portion of the retina must see the same object point. The muscular effort required to rotate each eye toward the point of convergence provides feedback which is a depth cue. This cue, called convergence, is rather weak and functions as a supplement to the most powerful physiological depth cue: disparity. When the eyes are converged on a given point, it appears as a single point. Objects in front or behind this depth appear as double images. The eye and brain are sensitive to the disparity of images at different depths and are thus able to interpret this disparity as difference in depth.

Convergence and disparity are not entirely separate. Both contribute to stereopsis and will be treated as a single cue. The lesser role of convergence as a cue to absolute depth may be separated from other aspects of stereopsis.

Most of the many monocular visual cues are considered psychological; only one (accommodation,) is physiological. One very powerful depth cue presented by simulators is motion parallax, or the relative visual flow of objects as one moves about. Nearby objects appear to move more rapidly while distant objects remain fixed. This cue also contributes greatly to the sensation of motion. Retinal image size, perspective, occultation, fog, haze, lighting, shade, and textural gradient all contribute to the three-dimensional appearance of simulated scenes. Accommodation, the focusing of the lens of the eye, provides physiological feedback from the muscular effort required to focus on a particular object.

The relative importance of stereopsis as a depth cue is still a subject of debate, and is discussed later in this paper. Several points, however, are clear. Stereopsis works in conjunction with other depth cues and is scaled to other cues, particularly perspective[1]. It becomes more important when monocular cues are scarce or ambiguous. For example: a small object that is near may appear exactly the same (from a monocular viewpoint) as a larger object farther away. A near object moving at the same speed as the observer appears the same (in terms of motion parallax) as a very distant stationary object. Stereopsis may be used to discern the reality in each case.

Creating Stereoscopic Cues With Image Generators

In stereo photography and workstation graphics, the interdependence of visual cues is exploited in order to magnify objects, exaggerate stereo effects, and accommodate a wide range of viewing positions. The same may be done in flight simulators to replicate the effect of certain sensors, however, for out-the-window systems, the whole visual system is geared toward producing an image that is visually correct for only a single viewer: the pilot. This corresponds to what is called in photography 'the orthoscopic condition'[2]. That is, the combination of CIG and display/projection variables produce an image that corresponds geometrically to the database and dynamic objects being simulated. To maintain this condition with a stereo CIG implies that all stereo parameters are fixed by the geometry of the display system with respect to the pilot. There are many methods used to create and display stereoscopic images. Several likely candidates for use in simulators are discussed below. They may be grouped into two general categories: Time-parallel methods present the stereo pair to both eyes simultaneously, which in general requires separate displays for each eye, or separate areas of a single display. Time-multiplexed methods allow use of a single display for both images, but not at the same time. Left and right images are alternately displayed, and each eye is allowed to view the display only while the image intended for that eye is being displayed.

Independent direct presentation is a time-parallel method, currently being used in simulators. This method is used with helmet mounted display systems. Image selection is done by displaying the stereo images immediately in front of each eye with displays that do not physically overlap.

Two electronic time-multiplexing methods used with video displays are active PLZT (lead lanthanum zirconate titanate) or LCS (liquid crystal shutter) glasses. The lenses alternate between states of transparency and opaqueness in synchronization with the images being displayed. The images must be alternated at a relatively high rate so that the image viewed by each eye is flicker free.

In a variation of LCS, the glasses worn are passive circular polarizers with right and left circular polarized lenses. An active polarizing 'screen' is placed over the display/projector which polarizes the emitted light circularly right, and then left, in alternating fields. The polarized light passes through the lens with like polarization, and is blocked by the lens with opposite polarization, allowing only one eye at a time to view the displayed image. This variation has the advantage of having lighter glasses, and leaving the head untethered.

If virtual image optics are used, parallel channel axes (PCA) may be used to create the required parallax. This requires that left and right channels have the same rotation, but are always offset horizontally (with respect to the head) by the interocular distance, nominally 2.5 inches.

To produce correct parallax on direct view displays or domes, a crossed channel axis (CCA) technique may be used. With CCA, the eyepoint coordinates fed to the CIG are separated horizontally by the interocular distance. The pitch and roll of the two channels are identical; however, the heading (with respect to the head) of each is 'toed in' by a small angle which is dependent on the distance from the eye to the display. This method is required to create proper parallax when a single display or projector is used to display the images for both eyes. CCA is necessary to produce positive parallax for objects in screen space (i.e. those objects which are farther away than the display). The correct parallax is also produced for objects nearer than, or at the same depth as the display.

Another time-parallel method is to place linear sheet polarizers over each of two projectors, with the axes of polarization at right angles to each other. The observer wears glasses consisting of sheet polarizers with the axis of each polarized lens parallel to polarization axis of one of the projectors. The light from each projector passes only through the lens with the parallel axis of polarization and is blocked by the other lens, allowing each image to be viewed only by the eye for which it is intended.

The analglyph technique is a time-parallel method using two images superimposed on single display. They are distinguished from each other by the complementary colors used for each image. The viewer wears glasses which have lenses that are the same colors as the images. This method precludes producing full color scenes, but has found use in some monochrome applications.

All of the methods described above require the user to wear a special viewing device. There have also been methods of image selection developed in which no special glasses or displays need be worn. These methods are known as autostereoscopic. They involve such devices as vibrating mirrors and lenticular lenses. In general, these methods do not lend themselves to stereo pairs as produced by image generators.

EXPERIMENT

Stereo technology has been applied successfully to computer graphics. It has gained acceptance in molecular modelling, mechanical CAD, medical imaging, meteorology, and other areas where visualization of scientific data is important. In the simulation world, the attitude toward stereopsis has generally been that it is not useful beyond distances of 25 to 50 feet and is not used in most flight tasks. Currently, there is some interest in stereopsis to enhance certain training tasks, and results of experiments by some visual scientists suggest that the threshold of stereopsis extends to thousands of feet.

The purpose of experiments performed by the author is to quantify the typical threshold of stereopsis using a CIG and a CRT display. The results show that in the simulated world, stereopsis, as an independent visual cue, is effective to distances over 300 feet.

An Evans & Sutherland ESIG-1000 CIG was used to produce static images on a 19" SRL model 2106 display. Stereo pair data files were created which consisted of 12 rectangles, arranged in 4 rows of three columns, with different parallax values.

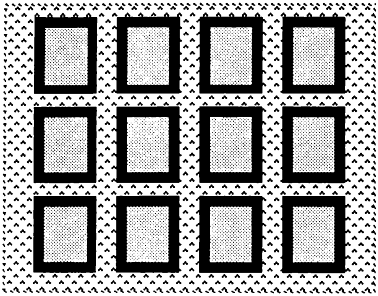


Figure 1. Test Pattern

These rectangles were surrounded by a planar grid. All of the rectangles were the same size in terms of pixels. There were 5 frames, each simulating the grid at different distances: 12, 18, 50, 200, and 1800 feet. All objects had positive (uncrossed) parallax. The parallax of some rectangles was greater than that of the grid, making them appear to be more distant than the grid. Others had a smaller value than the grid, causing them to appear closer than the grid. The difference in screen parallax values ranged from 10.85 arc minutes to 10 arc seconds. Special microcode was produced which had a field rate of 90 Hz. Time-multiplexed image selection was accomplished by means of an LCS visor and controller unit from Stereographics Corp. The controller was interfaced to the CIG and participants wore the visor to see the stereoscopic images. The resolution was 0.7 arc minutes per pixel, approximating the visual acuity in the center of the fovea.

Participants were asked to indicate for each polygon of each frame whether it appeared to be closer than, farther away than, or even with the depth of the grid. Twelve engineers consented to participate. Each frame viewed by each participant was scored based on the largest difference in parallax that was not identified correctly. The score given was the next greater value. For example, if a subject correctly identified all rectangles with values above 2.5 arc minutes, but missed the one with a value of 2.0 arc minutes, the figure used would be 2.5 arc minutes. On many frames, a subject missed a fairly high value rectangle, but successfully identified the relative positions of several lower value rectangles. These lower values (greater acuity) however, were not considered, in favor of a more conservative evaluation (e.g. even if the person in the earlier example correctly identified rectangles of 1.5 and 1.0 arc minutes, the score would still be 2.5 arc minutes).

Results

An average acuity level was calculated for each participant. The mean of these averages, which is used as a benchmark in this paper, is 2.3 arc minutes. This is a conservative figure, since the median of the best frames for each individual was 0.9 arc minutes. The standard deviation is 1.8 arc minutes. Three subjects (25%) received scores of 42 arc seconds (1 pixel) or better on two or more frames. Two of these individuals (16.7%) had one or more frames in which they were correct on every polygon, down to 10 arc seconds, and had averages of less than 1 arc minute.

Stereoacuity, Threshold of Stereopsis, and Distinguishable Depth.

Stereoscopic acuity, also called stereoacuity is the ability to detect a difference in depth between two objects based on stereoscopic cues. Visual scientists express stereoacuity in angular units. Like regular visual acuity (sharpness of vision), stereoacuity varies greatly from individual to individual. Unfortunately, however, there is no means of correcting poor stereoacuity. There are also many persons who are 'stereo blind'[2], just as there are persons who are color blind.

The threshold of stereopsis is distance beyond which the depth of an object is indistinguishable from that of an object at infinite depth (so far as stereopsis is concerned). In other words, the threshold is the maximum distance at which stereopsis is effective. There is a straight-forward geometrical relationship between stereoacuity and the threshold of stereopsis, as shown in figure 2a. This relationship is plotted in figure 3.

Two objects are distinguishable in depth by stereopsis if the difference of the convergence angles of the objects is greater than the stereoacuity of the individual viewing them. This relationship is illustrated in figure 2b. The experiments test the ability of the participants to distinguish differences in depth. Stereoacuity is evaluated to be the minimum difference in convergence angles that causes a perceived difference in depth.

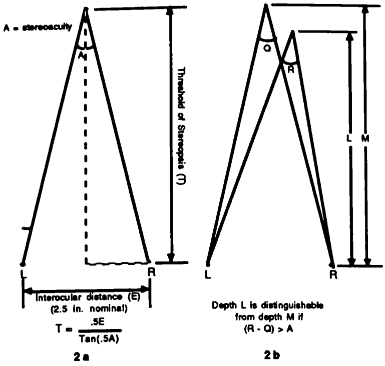


Figure 2. Acuity, Threshold, and Distinguishable Depth

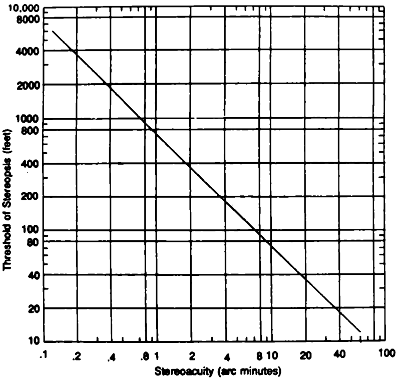


Figure 3. Stereoacuity vs. Threshold of Stereopsis

Stereoacuity of 2.3 arc minutes implies a threshold of stereopsis at 311 ft., as shown in figure 3. Stereoacuity of 42 arc seconds, as achieved by some members of the test group, implies a threshold of 1023 ft. Subjects with the poorest acuity, at 6.2 arc minutes, would have a threshold of greater than 100 ft., far more than the 50 ft. limit generally assumed. There is no doubt, however, that because of its geometry, stereopsis has a much greater effect with near objects. As shown in the curves of figure 4, for 2.3 minute acuity, an object at 100 ft. is barely distinguishable from an item at 75 ft., while at a nominal distance of 30 ft., a depth differential of 3 ft. can be discerned solely on the basis of stereoscopic cues (i.e. depths greater than 33 ft. or less than 27 ft.).

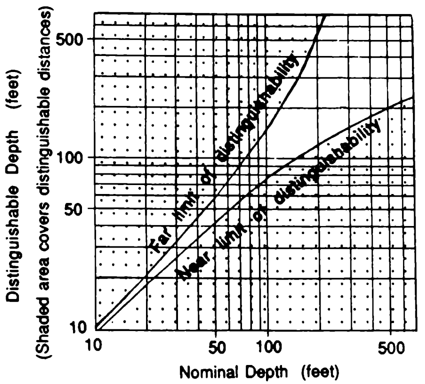


Figure 4. Distinguishable Depths With Stereoacuity of 2.3 Arc Minutes

An important conclusion from the experiments is that a significant percentage of people (17% in this case), were able to consistently detect and identify parallax differentials of less than a pixel. It should be noted that decreases in image brightness and contrast will negatively affect stereoacuity[3]. Modulation texture provides stereo cues, but not as effectively as polygon edges. The modulation is comparable to edges viewed with decreased visual acuity. There is evidence to suggest, however, that stereoacuity is largely unaffected by a decrease in visual acuity unless it is very severe[4].

One significant point not pertaining to stereoacuity is that although the images were fairly high brightness and high contrast, and at 45 Hz. per eye were close to the flicker threshold, flicker was not perceived by any of the participants even when it was suggested to them. It may be that this is due totally to the dimming effects of the LC shutters, but it might also indicate some interpolation between the eyes as far as flicker is concerned. Whatever the reason, it means that field rates less than 50 Hz. may be acceptable for producing flicker free, time-multiplexed stereo images, if the lower rate does not cause other unacceptable problems.

Stereopsis is generally considered as a cue to determine relative depth. i.e. the depth of one object vs. another. Experts do not agree on its role in perception of absolute depth. The experiments confirm the assertion that, in and of itself, stereopsis is not a good cue to absolute depth.

The 'range-finder' theory of convergence holds that by repeated experience, the effort required of the eye muscles to converge or fuse the two separate retinal images into a single image becomes a cue to the distance to the object in the same way a range finder is used in photography to find the distance to an object to be photographed[5].

The author's experiments discount the range finder theory. The subjects in the experiments were asked to estimate, if possible, the apparent distance to the grid. Virtually everyone complained about the difficulty of this task, and estimates that were given varied widely. However, when a reference was provided most participants were able to easily and accurately direct the reference to move closer or farther so as to place it at an equivalent depth with the virtual image being displayed. Experiments done by others agree with these results[2], leading to the conclusion that disparity plays a much greater role in stereopsis than convergence. In a simulator, as opposed to these experiments, there are familiar objects, the orthoscopic condition holds, all visual cues are coordinated, and stereopsis likely makes a greater contribution to determining absolute as well as relative depth.

Many subjects were not able to converge on the more distant patterns, particularly at 200 and 1800 ft. This, no doubt, is a result of the repetitive nature of the test pattern and the absence of monocular cues. It should be noted that many of the same individuals who could not converge the larger parallax values for the static test had no trouble converging on equally large parallax values in dynamic real-time scenes.

One common comment was about the ghosting (the viewing by the left eye of the image intended for the right and vice versa) which, for no apparent reason, was quite noticeable on the bottom row of rectangles, but hardly noticeable on the top two rows. Several commented that the task would have been easier if familiar objects were used, or if other cues such as perspective were present. Of course, this agreed with the objective of the test: to measure stereopsis in the absence of other cues.

Comparison To Results of Other Tests

Many stereopticians have done tests of stereoacuity. Their results generally give a much greater acuity than the experiment described in this paper. For example, in a study of 106 Soviets, 30% had stereoscopic acuity of better than 7 arc seconds[3]. Based on this value, the stereoscopic threshold for these people would be greater than a mile. Common tests used by optometrists to evaluate stereopsis test for acuity down to 40 arc seconds. One possible reason for the disparity in results is the difference in the method of the tests. Typical tests for stereoacuity require only an identification of difference in depth. The author's test went a step further by requiring correct interpretation of that difference. Thus, it might best be considered a measure of the 'cueing acuity', rather than merely sensitivity, as measured by many tests. It is this fact, and the fact that different sample groups were tested, that probably accounts for the lower values of acuity measured. Recent experiments at Honeywell, similar to the one described in this paper, obtained very similar results in terms of stereoacuity[6]. Researchers found a judgement error of 2.2 arc minutes in disparity detection.

Another possible limiting factor is that the resolution used, although greater than usual for simulators, is not fine enough to accurately render the extremely small angles indicated by some studies. Investigation into resolution constraints in the Honeywell experiments indicated that increased resolution would not significantly alter the results.

Cost and Benefits of Stereoscopic Flight Simulators

The requirement to calculate a separate view for each eye requires, basically, doubling the number of channels of CIG hardware or halving the field time. Not surprisingly, this factor alone has been enough to squelch most applications of stereo in simulators to date. New display systems must also be used, or old ones updated, to accommodate a higher field rate and lower persistence for time-multiplexing. For time-parallel methods, twice as many displays must be used. In addition, new software must be created in the form of real time system software to properly offset eyepoint coordinates. For time-multiplexing, microcode must be created for driving the higher field rate and multiplexing channels if required.

Although stereoscopic techniques and devices have been in use for many years, there are several technological advances in CIG technology and display systems that make the use of stereo in simulators more attractive and less costly today than in the past.

The introduction of large liquid crystal shutters is a great improvement over the PLZT material. PLZT shutters require high operating voltages and allow only about 6% of the available light to pass through in the 'transparent' state. LCS glasses require only low voltages and have a transmissivity of 30%, which has been subjectively described as similar to lightly tinted sun glasses. They have now proven themselves to be practical. Use of liquid crystal 'screens' offers the additional advantage of lightweight passive glasses.

The advent of helmet mounted display systems provides an obvious opportunity to utilize and test stereopsis as a visual cue. Image selection is provided by independent direct presentation, since separate displays for each eye are required. High resolution insets increase the effectiveness of stereopsis.

Software reconfigurability, such as that of ESIG-1000, can be a great cost saving advantage where stereo will be used in some, but not all, anticipated configurations. Rather than dedicating all the required hardware to producing stereo images, 'stereo-capable' systems may be built in which channels can be paired for stereo configurations and used separately for applications that may not require stereopsis. This adds yet another dimension of versatility to these systems.

Reconfigurability offers the additional capability to merge two channels of digital video data into a single video output with complete timing accuracy. This allows the multiplexing of two channels, each operating at a field rate of say 50 Hz. to be viewed from a single display with a 100 Hz. refresh rate, thus providing a flicker free 50 Hz refresh to each eye. This capability is essential to the production of time-multiplexed images which are flicker free without any degradation in scene complexity.

Slewable area-of-interest (AOI) projectors could also provide a cost savings in stereoscopic flight simulators since only the area of interest would need to be stereoscopic. Stereoacuity also rolls off sharply from the foveal region. Thus, only the area of interest would need to be generated and projected in stereo to obtain a fully stereoscopic simulator. This translates into less CIG and display hardware, and less cost than making all channels stereoscopic, while still providing a suitable display for the stereo images. Also, when the CIG receives head tracked eyepoint data, it eliminates parallax inaccuracies that occur when the direction of gaze is approximated to always be at the center of a fixed channel.

While the advances make stereo much more attractive than in the past, there are still many possible problems which must be considered for quality stereoscopic simulators to become a reality.

For example, when producing stereo images, it is possible to introduce many asymmetries, of which only one is desirable: the horizontal parallax which produces stereopsis. Intensity and color match, and raster size match must be carefully controlled in time-parallel systems. Time-multiplexed systems have the advantage that these parameters are inherently matched by virtue of using a single display. Time-multiplexing, however, possesses its own set of problems.

Due to the finite decay rate of CRT phosphors and projector oil films, ghosting may occur when left and right images are alternated rapidly, as required for flicker free operation. This effect is most noticeable on high contrast, high brightness polygon boundaries, and much less noticeable in textured scenes. The effects can be minimized if low persistence phosphors are used, or, in the case of light valve projectors, the oil temperature increases. This requires a tradeoff of less brightness to achieve the lower persistence.

Stereoscopic simulators are a step up in realism. Mayer and Cosman wrote, "The image produced by a CIG is, in truth, a very sophisticated illusion. The participant in a simulation exercise should feel, think, and react as if what he is seeing is real, when in fact it is not."[7] Just as virtual image optics are more believable because stereoscopic and accommodation cues make displayed objects appear to be a great distance away, stereoscopic simulators are more believable for objects at all distances, because stereoscopic cues accurately represent the real world.

An important reality that should be noted is that realism for the sake of realism is not the goal of simulation. The added realism must correspond to increased effectiveness. As Mayer and Cosman explained, "The effectiveness of the illusion is ... dependent on the system's ability to provide sufficient cues for a particular task." Some of the tasks for which stereo cues would be highly desirable, perhaps mandatory, are discussed below.

Naturally, the best applications for stereo are those where objects are viewed at relatively close distances, and/or where there is a lack of some of the previously mentioned depth cues. Both of these conditions apply to aerial refueling. Stercoptic cues could aid greatly in training this difficult task.

Formation flight could benefit from the addition of stereo cues for the same reasons as aerial refueling. With both of these tasks, there is a need to maintain a precise position with respect to another aircraft without the benefit of the constant motion parallax present in other maneuvers.

Low speed nap-of-earth simulation would also likely benefit from the presence of stereo cues. This task requires precise positioning of the helicopter with respect to fixed, as well as moving, close-range objects.

Space operations, such as docking and manipulator arm operation, exemplify an application where all available cues, static and dynamic, are needed to successfully perform these slow, precise, and short range maneuvers with accuracy measured in inches.

Conclusion

Stereopsis serves as an independent depth cue, and serves to enhance other cues. Although stereo simulators entail increased cost over monoscopic systems, advances in CIG and display technology lower the cost and the risk of stereoscopic systems while improving the quality of the stereo images. The results of the experiments described indicate that, in simulators, stereopsis is a valid cue at distances of up to 300 ft. The true value of stereo to simulation will require studies using pilots in real-time scenarios. As methods for producing high-quality stereo images are realized on simulators, these types of studies should and are being done. The future for stereo simulators appears promising. Utilizing clever architectural reconfigurability, stereo-capable systems such as the Simulator Complexity Test Bed for the U. S. Army, are now practical. There are several simulator applications in which stereopsis plays an important role. Stereoscopic simulators are currently being developed for some of these tasks. As methods for producing high-quality stereo images are realized on simulators, more effective training will result.

References

[1] MacAdam, D.L.; "Stereoscopic Perception of Size, Shape, Distance, and Direction"; Journal of the Society of Motion Picture and Television Engineers; 1954

[2] Lipton, Lenny; Foundations of the Stereoscopic Cinema; Van Nostrand Reinhold; New York; 1982

[3] Valyus, N. A.; Stereoscopy; Focal Press; London, New York; 1962.

[4] Welch, B.L.; "Characteristics of Flight Simulator Visual Systems"; Proceedings of the 4th Interservice/Industry Training Equipment Conference; Washington D.C.; 1982

[5] Spottiswoode, Raymond; and Spottiswoode, Nigel; The Theory of Stereoscopic Transmission and its Application to the Motion Picture; University of California Press; Berkeley; 1953.

[6] Yeh, Yei-Yu, and Silverstein, Louis D.;"Depth Discrimination in Stereoscopic Displays"; Society for Information Display Digest of Technical Papers; 1989

[7] Mayer, Neal L., and Cosman, Michael A. ; "Enhancing the Computer Generated Illusion"; Proceedings of the 4th Interservice/Industry Training Equipment Conference; Washington D.C.; 1982.

STEREOPSIS CUEING EFFECTS ON A SIMULATED PRECISION ROTORCRAFT "HOVER-IN TURBULENCE" TASK

89-3289-CP

Williams, S. P.; and Parrish, R. V.

Abstract

A real-time piloted simulation experiment of a precision rotorcraft "hover-in-turbulence" task was conducted to assess the efficacy of stereopsis depth cueing in pictorial flight displays. Six pilots endeavored to maintain a hover by visually aligning a set of inner and outer wickets (major elements of a "real-world" pictorial display), thus attaining the desired hover position, in a full-factorial experimental design. The display conditions examined included the presence or absence of a velocity display element (a velocity head-up display) as well as the stereopsis cueing conditions, which included non-stereo (binoptic or monoscopic - no depth cues other than those provided by a perspective, "real-world" display), stereo 3-D, and "hyper" stereo (telestereoscopic).

The objective results and subjective comments indicated that the depth cues provided by the stereo displays enhanced the situation awareness of the pilot and enabled improved hover performance to be achieved. The velocity display element also improved the hover performance, with the best hover performance being achieved with the combined use of stereo and the velocity display element. Pilot control input data revealed that less control action was required to attain the improved hover performance with the stereo displays.

Introduction

Current electronic display technology can provide high-fidelity, "real-world", pictorial displays under flicker-free conditions that incorporate true depth in the display elements. Advanced pictorial flight display concepts that embody 3-D images are being conceived and evaluated at various flight display research laboratories, including NASA Langley. Innovative concepts are sought that exploit the power of modern graphics display generators and stereopsis cueing, not only in enhancing the situation awareness provided by pictorial displays, but also in displays for the declutter of complex informational displays and in providing more effective alerting functions to the flight crew.

The intuitively-advantageous use of three-dimensional display of three-dimensional information, rather than the conventional two-dimensional display of such information, has been investigated for years within the flight display community.[1-7] These efforts have been particularly intense for helmet-mounted heads-up display applications, as stereopsis cueing is an almost natural byproduct of binocular helmet systems.[1-4] Additional investigations utilizing electronic shutters or polarized filters, rather than helmet optics, to present separate left and right eye views have also been conducted.[4-7] Most of these investigations have reported favorable subjective opinions concerning the value of stereopsis cueing, and, when objective data was presented, it generally demonstrated modest performance gains, or at least no degradations, in comparison with non-stereo displays.

The focus of the bulk of these investigations has been the stereoptic enhancement of situation awareness provided by the heads-up, out-the-window visual environment of the fighter or rotorcraft pilot. In most cases, the displays were autostereoscopic, with the viewing direction being slaved to the head movement of the subject pilot. The flight tasks have generally been either target acquisition/recognition tasks or complex flight maneuvers. Pilot/vehicle performance measures comparing non-stereo and stereo presentations in a highly structured experiment utilizing a realistic and demanding, but relatively simple, task are sparse. References 6-7 report results from a simple situation recognition task in a simulated transport aircraft application, but it was a non-real time study.

The goal of the effort reported herein was to quantitatively determine the efficacy of stereopsis cueing in enhancing the situational awareness of pilots conducting precision tasks in a simulator environment. Specifically, the study addressed the effects of stereopsis cueing in a "real-world" pictorial display for a precision rotorcraft "hover-in-turbulence" task. The display environment presented a pictorial out-the-window scene without autostereoscopsis (fixed head position).

Participating Pilots and Task

Six active duty and operationally-experienced Army helicopter pilots participated in this study. The task chosen for the evaluation was a precision "hover-in-turbulence" task. The pilots endeavored to fly to and maintain a hover by visually aligning sets of inner and outer wickets, as shown in a top view in Figure 1 and in a perspective view in Figure 2, thus attaining the desired hover position, in a full-factorial experimental design. Figures 2 and 3 are photographs of the display as viewed by the pilot. The task was initiated in a hover condition at a location displaced from the desired position in all three directions (250 feet longitudinally, 10 feet laterally, and 25 feet in altitude). The pilot was required to fly to the perceived position and reacquire a hover, with warning buzzers sounding at the one-minute-to-go, thirty-seconds-to-go, and begin-data-collection times. The performance metrics for the study included root mean square (RMS) values of the radial displacement from the desired hover point (radial error), as well as the pilot control inputs, taken for a period of one minute. If the pilot felt he had achieved the desired condition before the final buzzer indicating the start of data collection, he could initiate data collection at any time by closing the trigger switch in the cyclic controller.

The main factor of interest in the experiment was, of course, the display condition. The display conditions examined included non-stereo (no depth cues other than those provided by a perspective, "real-world" display, such as size, shape, interposition, motion parallax, etc.), stereo 3-D, and "hyper" stereo. The latter condition represented the case of the pilot's eyes (image sources) being located five feet apart (as might be encountered with Forward-Looking Infra-Red, FLIR, cameras mounted on each side of a cockpit for a binocular display), rather than the usual two and a half inches, and the condition exaggerated the depth cues present in the display.

The "real world" pictorial display (Figures 2 and 3) consisting of a ground grid, a sky/earth horizon, and the arrangement of wickets to allow visual alignment to determine the hover position, was chosen, rather than a flight director-type hover display (as in references 8-9, for example), because of the desire to investigate the stereoptic enhancement of the display in improving the situational awareness of pilots. The depth cues available in a synthetic heads-up, out-the-window visual environment seemed to lend themselves more naturally to such an investigation than any display consisting of symbology elements.

Another factor in the experimental design was the presence or absence of a velocity display element (a velocity head-up display within the display format), as shown in Figure 2. The cross of the element remained fixed, while the box moved vertically and laterally to represent fore/aft velocity and lateral velocity, respectively. Whenever the combined velocity components (airspeed, including vertical velocity) exceeded five knots in any direction, the box turned red. When airspeed exceeded ten knots, the box turned black. The use of this explicit velocity information display as a factor in the experiment allowed an examination of the coupling between velocity and stereopsis cueing.

Training was initiated with no turbulence and with the velocity element display "on" for each of the three non-stereo/stereo conditions. Training then progressed through the inclusion of turbulence for each condition to the removal of the velocity display element. The RMS radial error score was reported to the pilot following each trial. Each pilot achieved approximate asymptotic performance for the six experimental conditions before data collection was begun. Four replicates of each condition were obtained from each of the seven pilots.

Simulator Description

The simulator was assembled with the following elements: mathematical model, computer implementation, visual system hardware, graphics generation software, and simulator cockpit.

Mathematical Model

A six-degree-of-freedom total force and moment mathematical model of a teetering rotor helicopter, including a modified blade element rotor model, was used in the study, It was a modified model of a Huey-Cobra helicopter with a stability augmentation system tuned so that the rate command handling characteristics of an S-61 helicopter were closely duplicated. The development of the program of the model is documented,[10] as are various applications of the model.[11-14]

Turbulence was introduced into the math model through the addition of gust components to the body-axis longitudinal and lateral velocity variables. The level of the turbulence was considered to be moderate by the participating pilots.

Computer Implementation

Figure 4 illustrates the three-stage computer pipeline used for this study. The mathematical model of the helicopter and the simulation hardware drives were implemented on the Langley real-time simulation system. This system, consisting of a Control Data CYBER 175 computer and appropriate interface equipment, solved the programmed equations 31.25 times a second. The average time delay from input to output (1.5 times the sample period) was approximately 48 milliseconds. The graphics generation software resided within a Digital Equipment Corporation VAX 8650 and consisted of the necessary transformation equations and the graphics database for the display. The ADAGE RDS 3000 graphics computer only made calculations directly related to drawing the display. Utilizing this graphics architecture,[16] the graphics displays could be produced at an update rate of 20 Hz. However, the communications link between the CYBER and the VAX limited the total simulation update rate to 15 Hz.

Stereo Visual System Hardware

The stereo visual system hardware operated on the video signals supplied by the graphics generation system. These video signals presented a non-interlaced frame, 512 by 512 pixels in resolution, at 60 Hertz, consisting of both the left- and right-eye stereo pair images (see Figure 5). The stereo visual system hardware (Figure 6) separated the left- and right-eye scenes and presented each alternately, at 120 Hz, spread across the entire monitor screen (time-multiplexed stereo, resulting in a loss in vertical resolution of 50-percent), as shown in Figure 2. Liquid Crystal Device (LCD) glasses were shuttered in synchronization with the stereo pair, such that the right eye saw only the right eye scene and the left eye saw only the left eye scene, each at 60 Hz, without flicker. The stereo visual system hardware was developed by the Stereographics Corporation, Inc.[15]

Graphics Generation Software

Figure 7 illustrates the geometric principal that was employed to produce the left- and right-eye views within the stereo pair generation software. The heavy horizontal line represents the screen of the display monitor. To present an object that appeared at the depth of the screen, the object was drawn in the same location for both stereo pair views. For objects to appear behind the screen, the object was displaced to the left for the left-eye view and to the right for the right-eye view (with the displacement reaching a maximum value to place an object at infinity). For objects to appear in front of the screen, a displacement to the right was used for the left-eye view and to the left for the right-eye view.

To generate this lateral displacement, which is known as lateral disparity, left-eye and right-eye coordinate systems were offset from the viewer coordinate system of the visual scene. This offset was different for the various display conditions represented in the experiment. The non-stereo condition used an offset of 0, the stereo condition had an offset of 12 inches, and the hyper condition used an offset of 30 inches. Clipping was employed to limit each eye-view to the display surface boundaries. Asymmetric clipping, which provides an increased monocular field-of-view for each eye, with accompanying increases in the binocular fields-of-view, was implemented in the graphics software (see Figure 8).

Simple perspective division was used to transform the three-dimensional viewing volumes to two-dimensional viewports, whose centers were offset from the center of the display screen by half of the maximum-allowed lateral disparity (used to represent objects at infinite distance). Figure 9 represents the mapping of the visual scene to the stereo 3-D viewing volumes for the stereo and hyper stereo display condition cases.

Simulator Cockpit

The general-purpose fighter/helicopter cockpit of the Langley Visual Motion Simulator (VMS) was utilized in a fixed base mode for this study. The cyclic center-stick and the rudder pedals were loaded by a hydraulic system coupled with a special-purpose analog computer to provide realistic control forces. The collective stick is a counter-balanced, friction-controlled stick, and it is representative of a helicopter collective. No head-down instrumentation other than the primary display monitor was utilized. Because of structural limitations within the cockpit, the 19 inch monitor was mounted on the top of the instrument panel, approximately 19 inches from the pilot's eye position, rather than within the panel.

Experimental Results and Discussion

The investigation was designed as a full-factorial, within-subjects experiment, with pilots, display conditions, velocity display element, and replicates as the factors. The objective results are presented and discussed first, with the subjective results discussed thereafter.

Analysis of Objective Results

The data collected in the full-factorial experiment were analyzed using univariate analyses of variance for each metric. Table 1 is a summary of the results of these analyses for the five performance measures. The presentation of the results follows the statistically significant sources of variance identified in the table. Each of the main factors of the experiment are discussed, including any relevant analyses of associated interaction terms, for the main performance measure of interest, radial error, and the four control inputs. The Display by Velocity Display Element interaction is also discussed in these terms within both main effects (as D × V for the Display Condition and as V × D for the Velocity Element factors - D × V and V × D are the same interaction term).

Pilots.- The main effect of pilot variability was highly significant for all performance measures. This result is always expected in a precision task, and the pilot variability was therefore isolated from the rest of the analyses by its inclusion as a main factor in the experiment.

Display condition.- The analyses revealed that the main effect of display condition was highly significant for every measure, with the exception of the RMS pitch input. The pitch input activity did not change as a function of display condition. This result was somewhat surprising, since pitch control is used to maintain fore/aft position with respect to the desired hover point, and one might expect any depth cueing effects to be realized along the fore/aft axis (the "depth" axis). However, none of the other factors in the analysis of the RMS pitch results was significant either, with the exception of variablility between pilots. The handling characteristics of the simulated vehicle were such that the lateral control task was much more difficult than longitudinal control. However, this fact is offered as a suggestion rather than as an explanation for this result.

The other measures did change as stereopsis cues were added to the display. Table 2 provides a synopsis of the results in terms of percent reductions from the non-stereo display condition for each of the measures. For the four measures that show changes, the RMS level for the non-stereo condition was significantly greater than that of either stereo condition. There were no differences detected between the stereo and hyper stereo performances for three of these measures. The addition of stereopsis cueing to the display resulted in a reduction from the non-stereo level of about 28-percent for the radial performance error (Figure 10), 16-percent for the roll activity (Figure 11), and 10-percent for the collective activity (Figure 12). The RMS roll and RMS collective reductions were not consistent across all of the pilots (as indicated by the significance of the pilot by display condition interaction), while the radial error reduction effect was consistent.

For the fourth measure, RMS pedal input, differences were detected between the stereo and hyper stereo performances, both of which were significantly less than the non-stereo performance (Figure 13). The hyper stereo activity was significantly less than the stereo condition (about a 10-percent reduction from the stereo condition), and the stereo input activity was less than the non-stereo input activity (about an 8-percent reduction). This reduction was the only difference detected in the objective data for the main effects between the stereo and hyper stereo conditions. The RMS pedal reductions were not consistent across all of the pilots. One explanation for this result is that, with the closer appearance of the wickets with the hyper stereo presentation, perhaps some of the pilots could detect a directional error earlier and thus required a smaller RMS pedal input to ensure correction.

These results are considered to indicate that the depth cues provided by the stereo displays enhanced the situation awareness of the pilot and enabled improved hover performance to be achieved, with less control action required to attain the improved hover performance.

Display condition by velocity display element interaction.- Table 3 provides a synopsis of the results for this second order interaction in terms of percent reductions from the non-stereo display condition for each velocity element display condition for all measures. This interaction, D by V, was not significant for the radial performance masure, indicating that the effect of display condition on the radial performance error does not vary with the presence or absence of the velocity element display.

Indeed, Newman-Keuls t-test comparisons[17] between the means for each display condition within each velocity element condition (the means are shown in Figure 14) revealed that the non-stereo performance error was significantly greater than that associated with either stereo condition. There were no statistically significant differences detected between the stereo and hyper stereo performance errors for either velocity element condition. The addition of stereopsis cueing to the display resulted in a reduction of about 26-percent from the non-stereo performance error for the velocity element "on" condition, with a 29-percent reduction for the velocity "off" condition. Thus, the display condition effect was almost constant, regardless of the presence/absence of the velocity element display.

The second order interaction (D × V) was significant for three of the input measures. Figures 15-17 show the mean RMS inputs for roll, pedal, and collective activities, respectively. From Table 3, it is evident that for the velocity element display "off" condition, that stereopsis cues enabled improved hover performance with less control activity. However, for the velocity element display "on" condition, superior performance was achieved with the same level of input control for the stereo display condition. For the hyper stereo condition, superior performance was achieved with somewhat reduced input levels, although RMS collective input levels remained constant.

The stereopsis enhancement was particularly effective in reducing control activity when the velocity information element was absent from the display, implying that some velocity information, as well as positional information, can be readily extracted from the depth presentation. With the velocity information already provided by the velocity element display, the anticipated reduction in control activity with the addition of stereopsis cues is perhaps no longer available.

Velocity display element "on/off".- One would expect that the direct display of velocity information would result in improved performance, with lower control activity, regardless of the display condition (i.e., the V main effect would be significant and the V × D interaction term would not be). The analyses revealed that the main effect of velocity display element "on/off" was highly significant only for the radial performance error and pedal activity measures. Table 4 provides a synopsis of the results in terms of percent reductions from the velocity element display "off" condition for each of the measures. There was a 34-percent reduction in the RMS radial error (Figure 18) and a 22-percent reduction in the pedal activity measure (Figure 19) when the velocity display element was presented to the pilots. However, these reductions were not consistent across all pilots (as indicated by the significance of the pilot by display condition interaction).

Velocity display element by display condition interaction.- This second order interaction (V × D) was significant for three of the input measures (roll, pedal, and collective activities - Figures 15-17, respectively). Table 5 provides a synopsis of the results in terms of percent changes from the velocity element display "off" condition for each display condition for all measures. As stated previously, one would expect that the direct display of velocity information would result in improved performance, with lower control activity, regardless of the display condition (i.e., the V main effect would be significant and the D × V interaction term would not be). And this expectation was realized for the radial error and RMS pedal input measures. Examining the velocity element effectiveness across display conditions from the data of Table 5 reveals a reduction of 36-percent with the addition of the velocity display element to the non-stereo display, a 27-percent reduction with the stereo display, and a 38-percent reduction with the hyper stereo display. These differences were not statistically significant, and therefore the velocity element effect was almost constant, regardless of the display condition being utilized.

The amount of reduction in pedal activity did vary with display condition (the D × V interaction term was significant). For the non-stereo condition, a reduction of 30-percent occurred in pedal activity, while for the stereo and hyper stereo conditions, the reductions were 8- and 26-percent, respectively.

For the roll and collective activity measures, the V main effect was not significant and the D × V interaction term was significant. The expectation that the direct display of velocity information would result in reduced activity levels was met for the non-stereo condition with both the RMS roll and RMS collective inputs, with a reduction in both cases of 12-percent. However, for these two measures, there was no significant reduction in activity for the hyper stereo condition when the velocity element was "on", and for the stereo condition, there were actually increases in activity (a 23-percent increase in roll and a 9-percent increase in collective activity). The reductions and increases acted to cancel the significance of the main effect for these two measures.

In summary, these results are considered to indicate that the direct addition of velocity information provided by the velocity element display enhanced the situation awareness of the pilots and enabled improved hover performance to be achieved consistently across all display conditions. However, the lack of a consistent effect for the addition of velocity information on control input activity was unexpected. The expected effect was realized for the non-stereo display condition for most of the input measures (reduced activity with the addition of velocity information). For the hyper stereo condition, the expected effect was realized for one measure (pedal activity), with no effect occurring for the other measures. The stereo condition situation is very ambiguous, with control activity decreasing, increasing, and remaining the same, depending upon the particular measure, with the addition of the velocity display. No explanation is offered for this result.

The fact that the expected effect of velocity cues on control activity was realized for the non-stereo condition but not for the stereopsis conditions suggests that some differences exist in the velocity information inherently imparted by the addition of stereo depth cues.

Replicates.- The main effect of replicates was significant for only the RMS pedal and RMS collective measures. In both cases, there was no difference between the levels of replicates 1 and 2. For the RMS pedal measure (Figure 20), there was a reduction of 13-percent between the average of 1 and 2 and the average of 3 and 4 (with no difference detectable between 3 and 4). For the RMS collective measure (Figure 21), there was a significant reduction of 8.5-percent between the average of 1 and 2 and the level of replicate 3. There was an additional reduction of 9-percent between the level of 3 and the level of replicate 4.

These results are not really surprising, as the overall direction of reductions in control activity with increasing replications is a classical pattern associated with learning a task.

Subjective Results

Unstructured pilot comments recorded throughout the experiment indicated that every pilot preferred the stereo display condition. They felt that they were aware of where they were relative to the desired hover point, and that they could detect upsets from the turbulence earlier and make the necessary corrections more readily with the stereo display condition. The non-stereo display sometimes became just a conglomeration of lines that required mental sorting to achieve situation awareness. The hyper stereo condition was not liked by most of the pilots, although they acknowledged that the display provided the same increased situation awareness that the stereo display provided. The dislike was attributed by some pilots to the situation sometimes encountered upon close approach to one of the front wickets. The exaggerated depth of the hyper stereo display appeared to allow the front wicket to penetrate into the cockpit with the pilot, a situation that they found to be very distracting.

Concluding Remarks

The objective and subjective results of this experiment indicate that stereopsis cueing is a most effective way to enhance the situation awareness of pilots utilizing pictorial displays. The depth cues provided by the stereo displays enhanced the situation awareness and enabled improved hover performance to be achieved. Control input measurement data revealed that less control action was required to attain the improved hover performance with the stereo displays. The stereopsis enhancement was particularly effective (resulted in reduced control activity) when the velocity information element was absent from the display, implying that some velocity information, as well as positional information, can be readily extracted from the depth presentation. With the velocity information already provided by the velocity element display, the anticipated reduction in control activity with the addition of stereopsis cues is perhaps no longer available.

The results also indicate that the direct addition of velocity information provided by the velocity element display dramatically increased the situation awareness of the pilots and enabled improved hover performance to be achieved. However, the lack of a consistent effect for the addition of velocity information on control input activity was unexpected, and no explanation is offered for this result, although it is suggested that some differences exist in the velocity information inherently imparted by the addition of stereo depth cues.

The best hover performance was achieved with the combined use of stereo depth cues and the velocity display element.

References

1. Mountford, S. J.; and Somberg, B.: Potential Uses of Two Types of Stereographic Display Systems in the Airborne Fire Control Environment. Proc. of the Human Factcrs Society, 1981, p. 235-239.

2. Setterholm, J. M.; Mountford, S. J.; and Turner, P. N.: Assessment of Stereographics for Fire Control and Navigation in Fighter Aircraft. AFWAL-TR-82-3008, 1982.

3. Woodruff, R. R.; Hubbard, D. C.; and Shaw, A.: Advanced Simulator for Pilot Training and Helmet-mounted Visual Display Configuration Comparisons. AFHRL-TR-84-65, 1985.

4. Kruk, R. V.; and Longridge, T. M.: Binocular Overlap in a Fiber Optic Helmet Mounted Display. The IMAGE 3 Conference Proc., 1985, p. 363-378.

5. Kim, W. S.; Ellis, S. R.; Tyler, M. E.; and Hannaford, B.: Quantitative Evaluation of Perspective and Stereoscopic Displays in Three-Axis Manual Tracking Tasks. IEEE Transactions on Systems, Man, and Cybernetics, Vol. SMC-17, No. 1, Jan./Feb., 1987.

6. Nataupsky, M.: Piloted Evaluation of Stereo and Non-Stereo Pictorial Displays for Curved Approach to Landing. Spatial Displays and Spatial Instruments Conference and Workshop, Aug. 1987.

7. Nataupsky, M.; and Crittenden, L.: Stereo 3-D and Non-Stereo Presentations of a Computer-Generated Pictorial Primary Flight Display with Pathway Augmentation. Proc. of the AIAA/IEEE 8th Digital Avionics Systems Conference, Oct. 1988.

8. Eshow, M. M.; Aiken, E. W.; and Hindson, W. S.: Preliminary Results of a Flight Investigation of Rotorcraft Control and Display Laws for Hover. American Helicopter Society Mideast Regional National Specilaists' Meeting in Rotorcraft Flight Controls and Avionics, Oct. 1987.

9. Hess, R. A.; and Gorder, P. J.: Design and Evaluation of a Cockpit Display for Hovering Flight. AIAA/AHS/ASEE Aircraft Design, Systems & Operations Meeting, Sept., 1988.

10. Houck, J. A.; Gibson, L. H.; and Steinmetz, G. G.: A Real-Time Digital Computer Program for the Simulation of a Single-Rotor Helicopter. NASA TM X2872, 1974.

11. Callan W. M.; Houck, J. A.; and Dicarlo, D. J.: Simulation Study of Intracity Helicopter Operations under Instrument Conditions to Category I Minimums. NASA TN D-7786, 1974.

12. Parrish, R. V.; Houck, J. A.; and Martin, D. J., Jr.: Empirical Comparison of a Fixed Base and a Moving Base Simulation of a Helicopter Engaged in Visually Conducted Slalom Runs. NASA TN D-8424, 1977.

13. Parrish, R. V.: Preliminary Investigation of Motion Requirements for the Simulation of Helicopter Hover Tasks. NASA TM 81801, 1980.

14. Ricard, G. L.; Parrish, R. V.; Ashworth, B. R.; and Wells, M. D.: The Effects of Various Fidelity Factors on Simulated Helicopter Hover. NAVTRAEQUIPCEN IH-321, 1981.

15. Model GDC-2 Graphics Display Controller Manual, Stereographics Corporation, San Rafael, CA., Feb., 1986.

16. Williams, Steven P.: A Graphics Architecture for Improved Performance in the Generation of Stereo 3-D Pictorial Displays. Proposed NASA TM, 1989.

17. Steel, R. G. D.; and Torrie, J. H.: Principles and Procedures of Statistics. McGraw-Hill Book Co., Inc., New York, 1960.

TABLE 1.- SUMMARY OF ANALYSES OF VARIANCE

FACTORS	Degrees of Freedom	RMS PERFORMANCE MEASURES				
		RADIAL	PITCH	ROLL	PEDAL	COLLECTIVE
PILOT, P	5	**	**	**	**	**
DISPLAY CONDITION, D	2	**	–	**	**	**
P x D	10	–	–	**	**	**
VELOCITY ELEMENT, V	1	**	–	–	**	–
P x V	5	*	–	–	**	–
D x V	2	–	–	**	*	*
P x D x V	10	–	–	–	–	–
REPLICATES, R	3	–	–	–	*	**
ERROR	105					

– not significant at levels considered.

* significant at 5-percent level.

** significant at 1-percent level.

TABLE 2.- SYNOPSIS OF DISPLAY CONDITION EFFECTS AS PERCENT REDUCTIONS FROM NON-STEREO LEVELS

DISPLAY CONDITION	RADIAL	PITCH	ROLL	PEDAL	COLLECTIVE
NON-STEREO	standard	standard	standard	standard	standard
STEREO	– 28	0	– 16	– 8	– 10
HYPER STEREO	– 28	0	– 16	– 18	– 10

TABLE 3.- SYNOPSIS OF DISPLAY CONDITION EFFECTS ACROSS VELOCITY DISPLAY ELEMENT ON/OFF CONDITION AS PERCENT REDUCTIONS FROM NON-STEREO CONDITION LEVELS

VELOCITY ELEMENT DISPLAY	DISPLAY CONDITION	RADIAL	PITCH	ROLL	PEDAL	COLLECTIVE
OFF	NON-STEREO	standard	standard	standard	standard	standard
	STEREO	- 29	0	- 26	- 19	- 18
	HYPER STEREO	- 29	0	- 26	- 19	- 18
ON	NON-STEREO	standard	standard	standard	standard	standard
	STEREO	- 26	0	0	0	0
	HYPER STEREO	- 26	0	- 12	- 18	0

TABLE 4.- SYNOPSIS OF VELOCITY ELEMENT EFFECTS AS PERCENT REDUCTIONS FROM ELEMENT OFF LEVELS

VELOCITY ELEMENT	RADIAL	PITCH	ROLL	PEDAL	COLLECTIVE
OFF	standard	standard	standard	standard	standard
ON	- 34	0	0	- 22	0

TABLE 5.- SYNOPSIS OF VELOCITY ELEMENT EFFECTS ACROSS DISPLAY CONDITIONS AS PERCENT CHANGES FROM ELEMENT OFF CONDITION LEVELS

DISPLAY CONDITION	VELOCITY ELEMENT ELEMENT	RADIAL	PITCH	ROLL	PEDAL	COLLECTIVE
NON-STEREO	OFF	standard	standard	standard	standard	standard
	ON	- 36	0	- 12	- 30	- 12
STEREO	OFF	standard	standard	standard	standard	standard
	ON	- 27	0	+ 23	- 8	+ 9
HYPER STEREO	OFF	standard	standard	standard	standard	standard
	ON	- 38	0	+ 4	- 26	+ 2

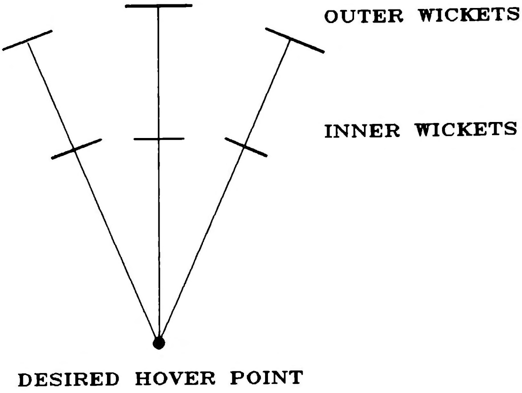


Fig. 1 Top view of rotorcraft precision hover task.

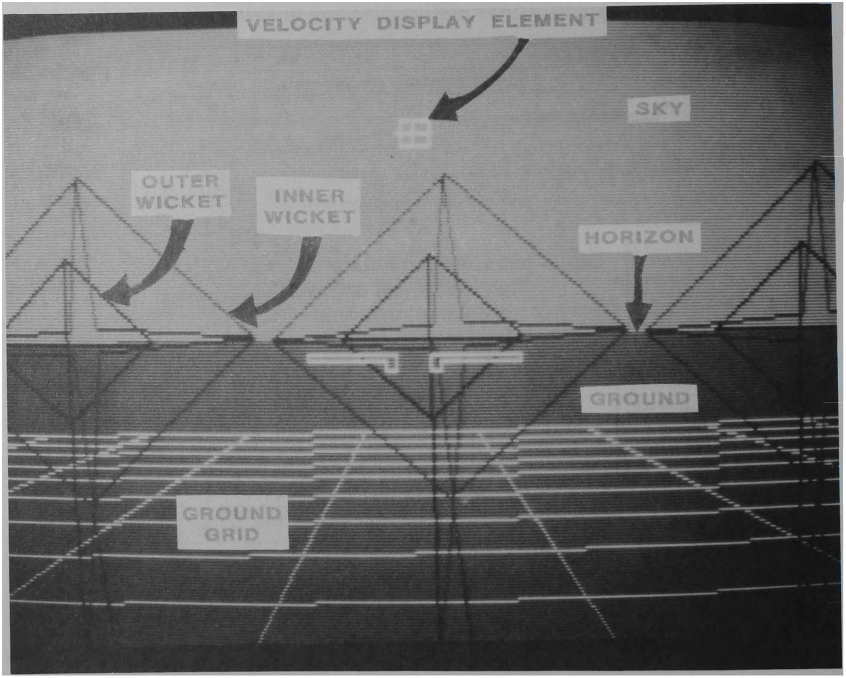


Fig. 2 Pilot's view of "real-world" perspective display when vehicle is near desired hover point.

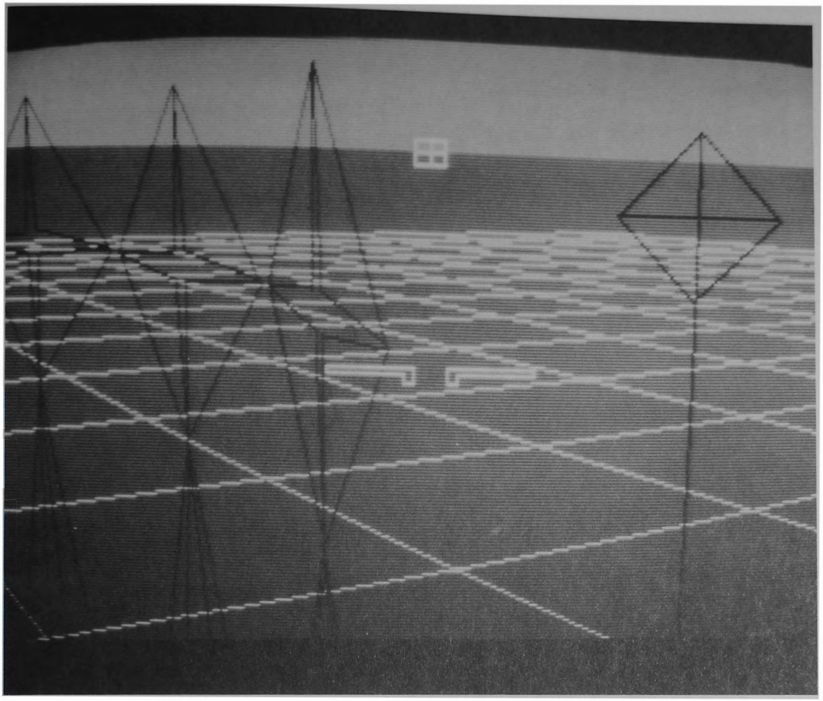


Fig. 3 Pilot's view of "real-world" perspective display when vehicle is right of desired hover point.

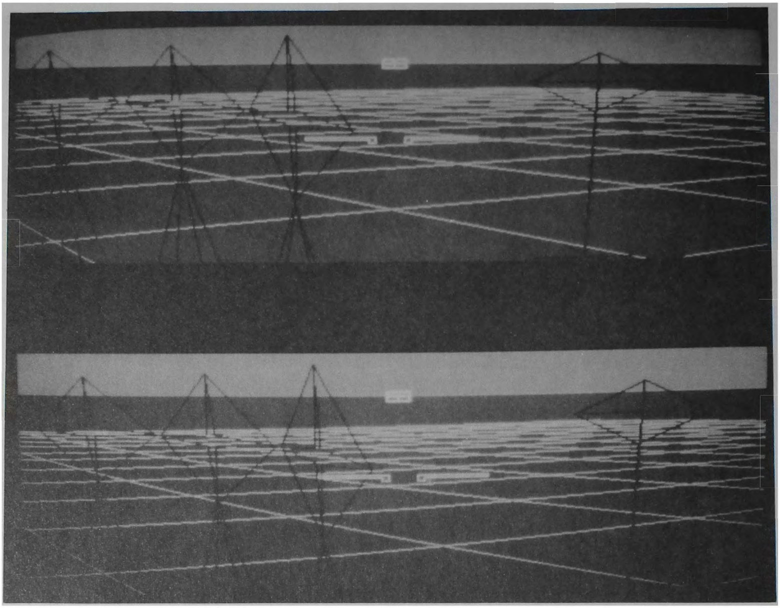


Fig. 4 Stereo pair images generated by the graphics generation system.

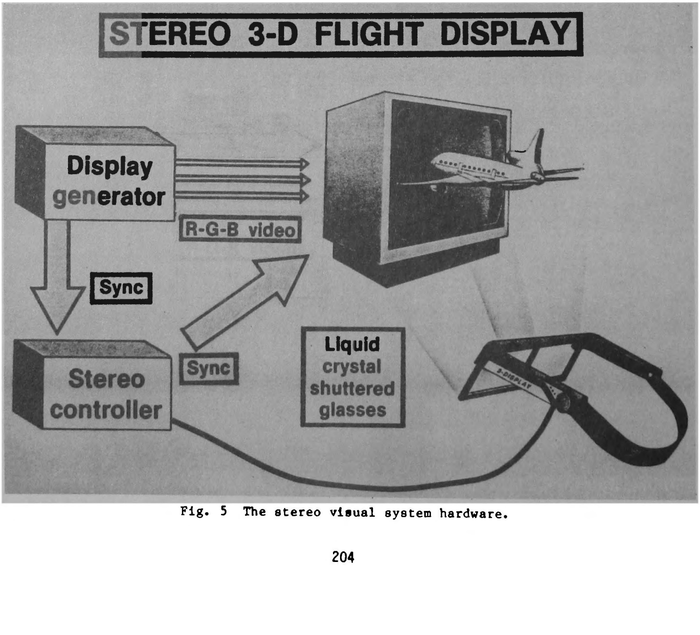


Fig. 5 The stereo visual system hardware.

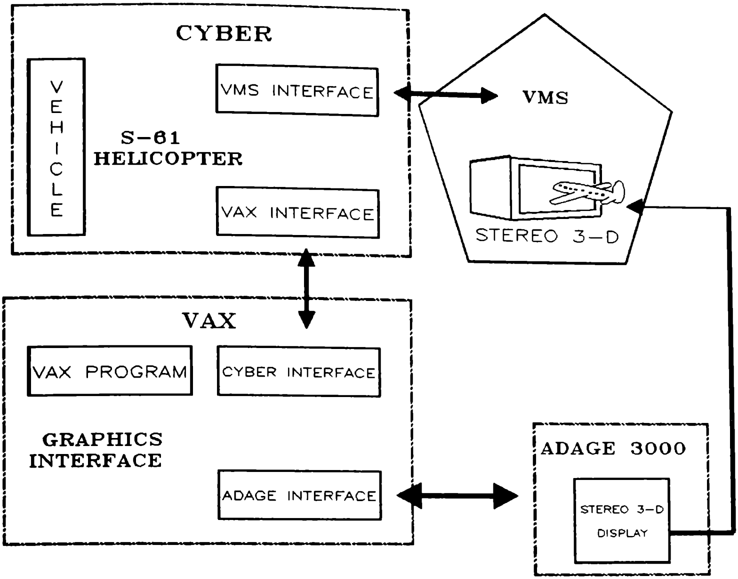


Fig. 6 The three-staged computer pipeline.

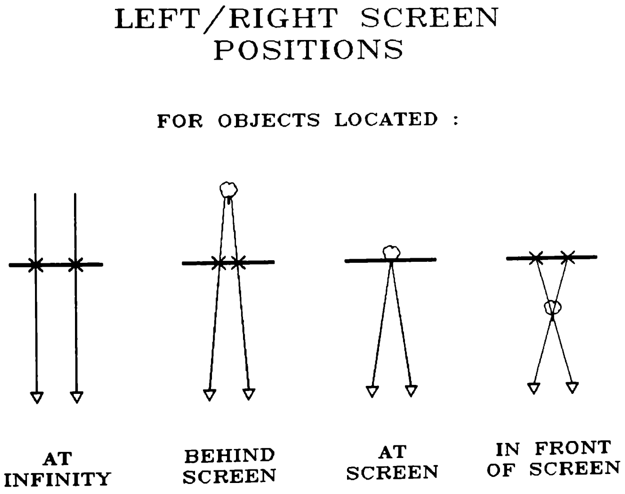
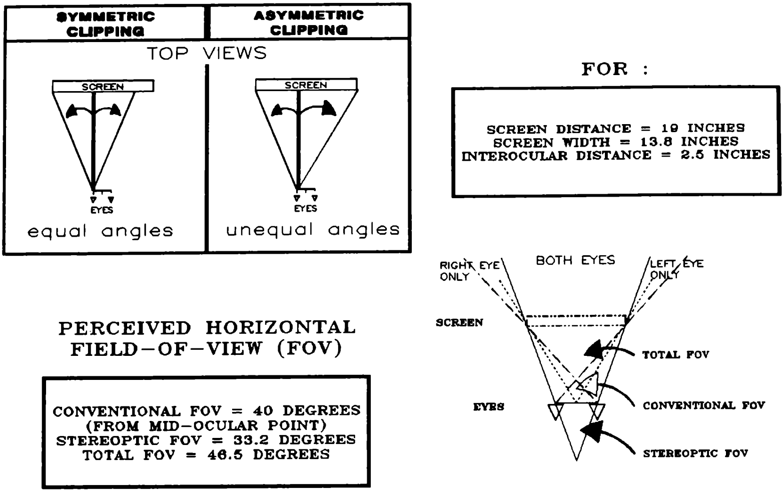


Fig. 7 The geometric principal for producing left- and right-eye views.



SCREEN DISTANCE = 19 INCHES
SCREEN WIDTH = 13.8 INCHES
INTEROCULAR DISTANCE = 2.5 INCHES

CONVENTIONAL FOV = 40 DEGREES
(FROM MID-OCULAR POINT)
STEREOPTIC FOV = 33.2 DEGREES
TOTAL FOV = 46.5 DEGREES

Fig. 8 Asymmetric clipping algorithm.

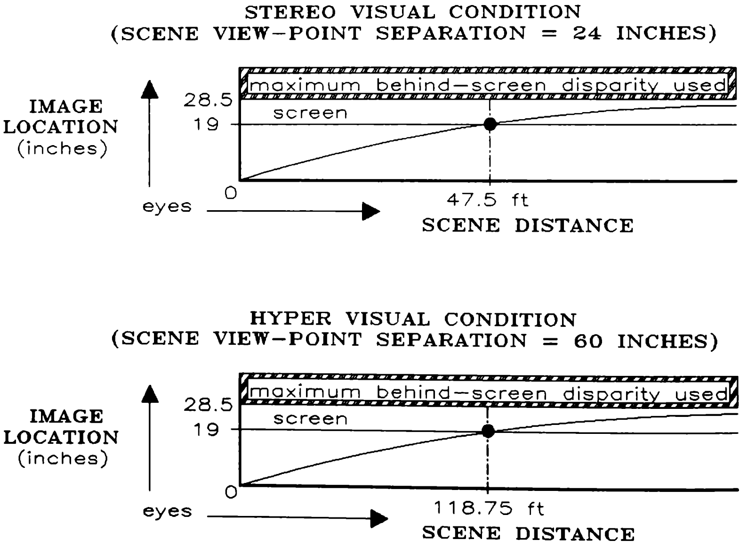


Fig. 9 Visual scene mapping to stereo 3-D viewing volumes.

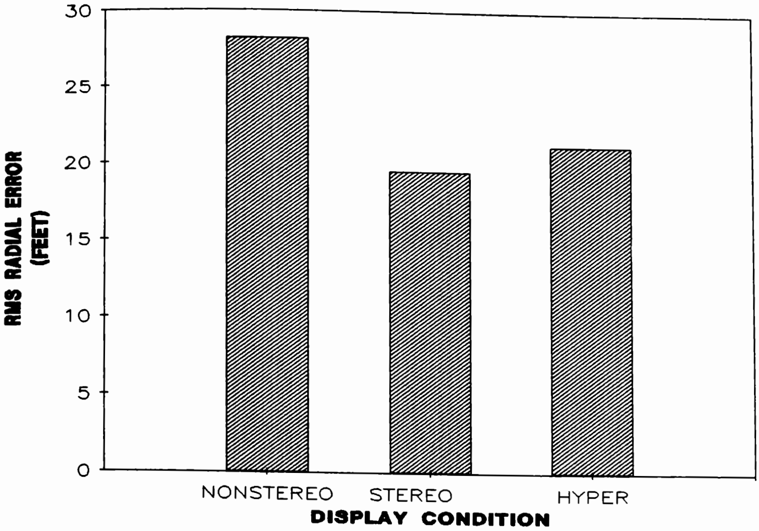


Fig. 10 Average RMS radial error for each display condition.

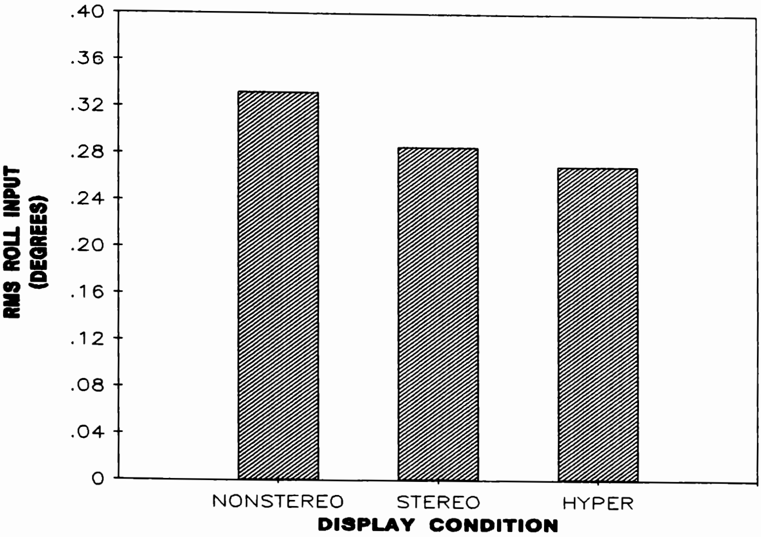


Fig. 11 Average RMS roll input for each display condition.

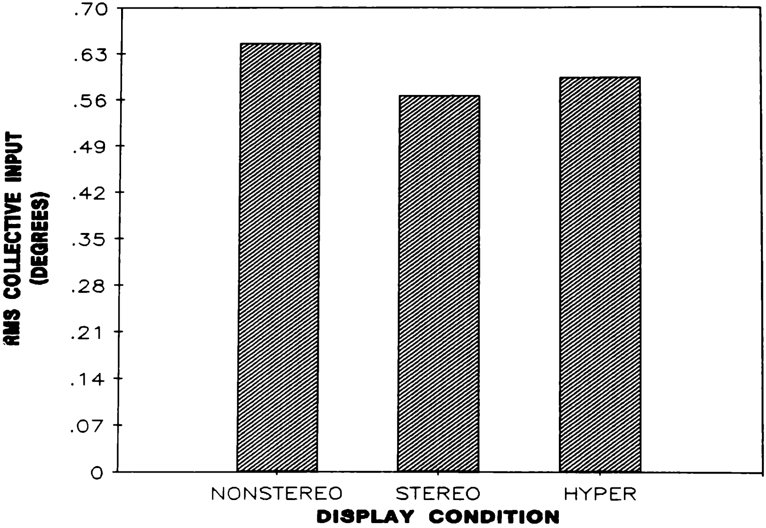


Fig. 12 Average RMS collective input for each display condition.

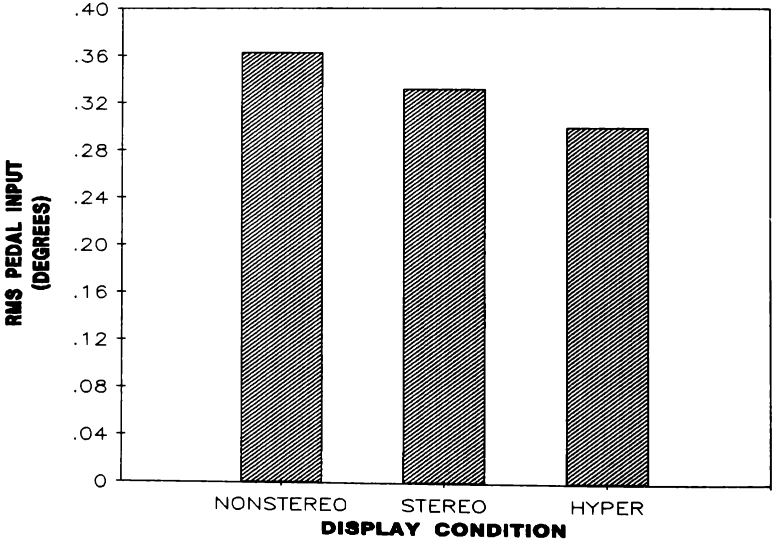


Fig. 13 Average RMS pedal input for each display condition.

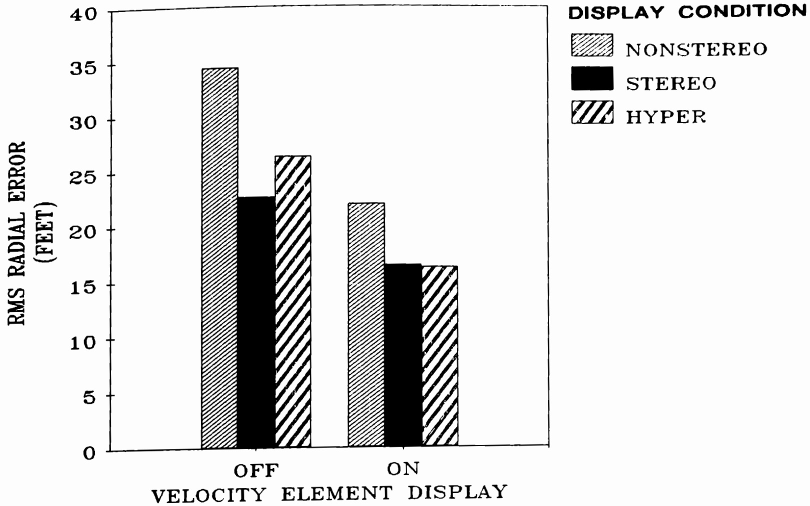


Fig. 14 Average RMS radial error for each display condition with the velocity element display on/off.

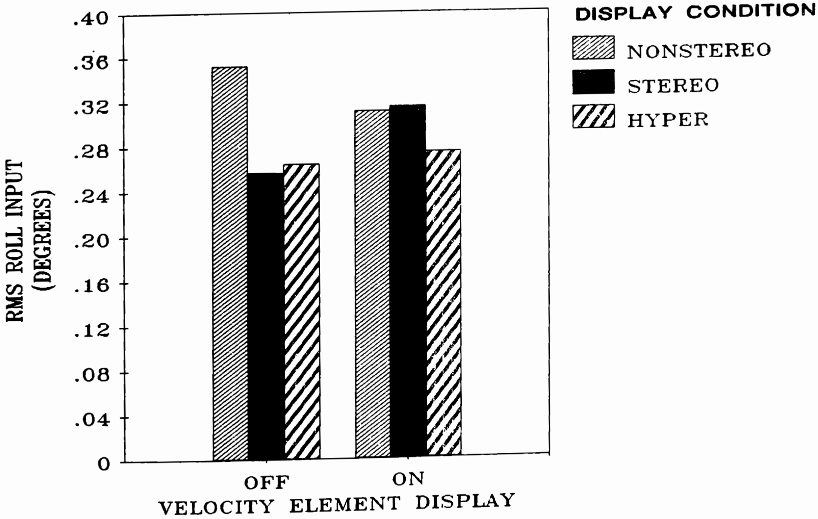


Fig. 15 Average RMS roll input for each display condition with the velocity element display on/off.

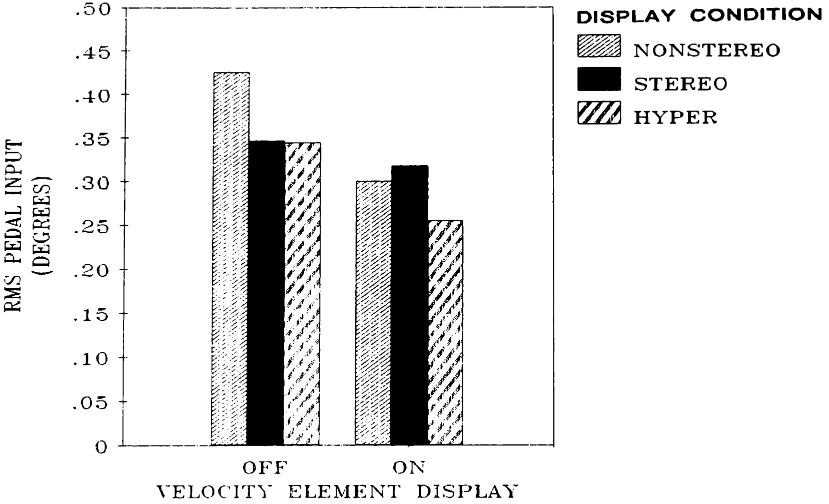


Fig. 16 Average RMS pedal input for each display condition with the velocity element display on/off.

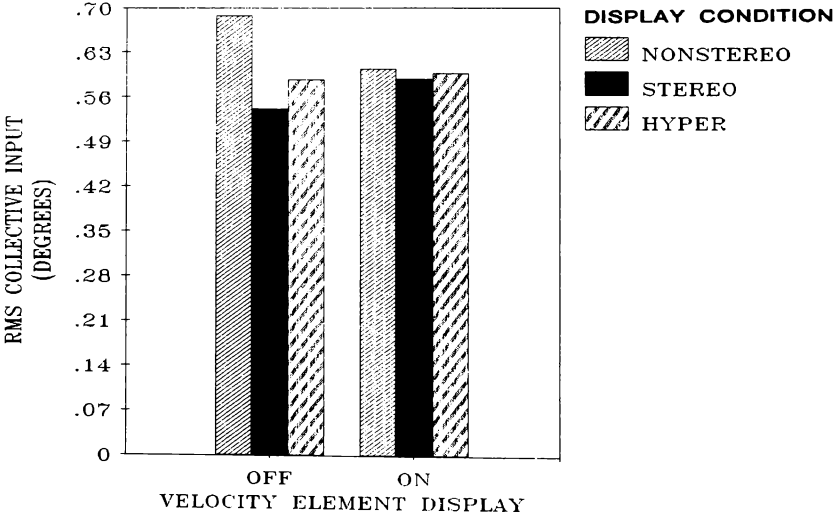


Fig. 17 Average RMS collective input for each display condition with the velocity element display on/off.

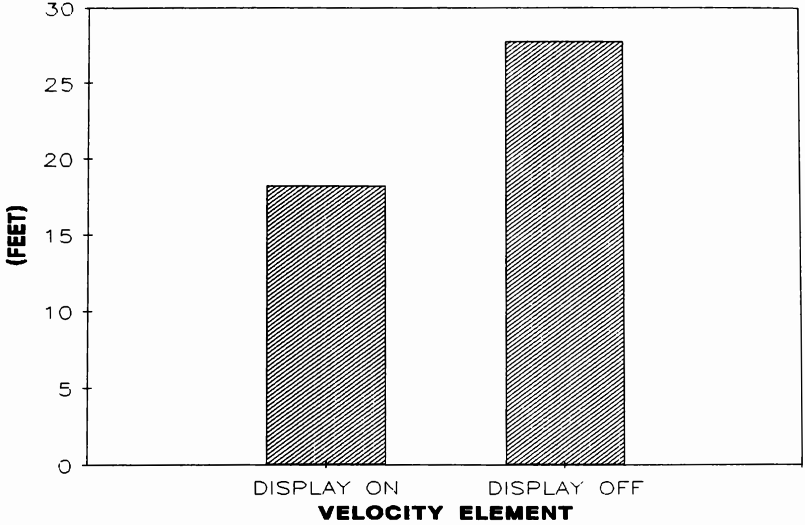


Fig. 18 Average RMS radial error with the velocity element display on/off.

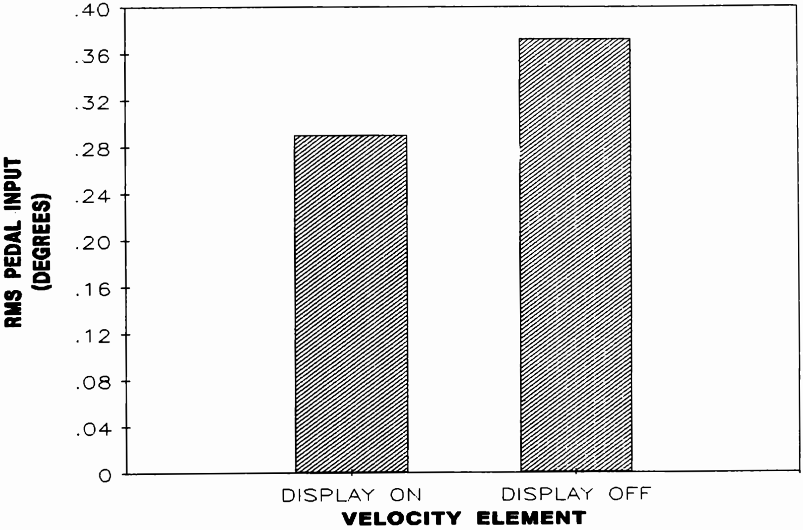


Fig. 19 Average RMS pedal input with the velocity element display on/off.

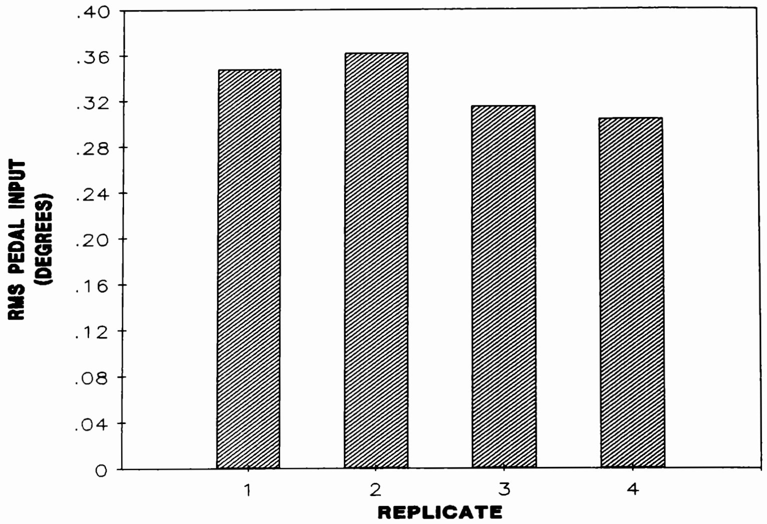


Fig. 20 Average RMS pedal input for each replicate.

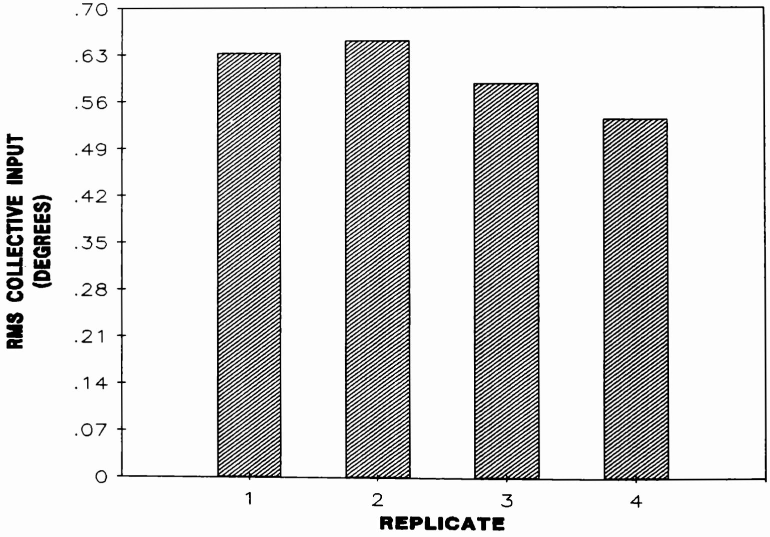


Fig. 21 Average RMS collective input for each replicate.

MODEL-BASED TERRAIN-FOLLOWING DISPLAY DESIGN

89-3290-CP

Paul G. Gonsalves*, Edward W. Kneller**, and Greg L. Zacharias***
Charles River Analytics Inc.
Cambridge, MA

Abstract

A model-based method for terrain-following cockpit display design and evaluation is presented. Two basic display configurations along with several display enhancements are evaluated via a combined analytical modeling and experimental simulation effort. The method entails the use of an integrated pilot/vehicle/display model to analyze and predict pilot performance trends as a function of display content/format. Model-based display design is supported and evaluated via real-time pilot-in-the-loop simulation.

Introduction

Current design methods for low-level terrain-following (TF) cockpit displays rely on past design experience, current engineering judgement, and extensive simulation evaluation. These methods are cumbersome and prone to uncertainty in the face of the continuing evolution of aircraft mission profiles and objectives. What is required is a more systematic approach to display design and evaluation that minimizes subjectiveness and accounts for the capabilities and limitations of the pilot. Such a method should be adaptive to a wide variety of display and vehicle configurations.

As the potential basis for such a systematic approach, we describe here a model-based method for display design and evaluation. The basic approach centers on the use of a pilot/vehicle/display model based on the Optimal Control Model (OCM), which combines general knowledge of human perception and performance, with specific knowledge of terrain-following aircraft and avionics capabilities. We describe the model, its use in the design of a man-in-the-loop terrain-following simulation, and its use in model-based analysis of the resulting flight simulation data. A number of candidate display aids are evaluated, including terrain preview, flight path command and pitch command directors, path predictors, and pictorial guidance displays. Model-based analysis of the resulting data is conducted to assess, within a model-framework, the relative contributions and effectiveness of the various display components.

Earlier attempts to develop a systematic approach to display design and evaluation have focused on specific display questions. Hess[1] defined and illustrated a detailed method for the analysis of instrument-type displays. The approach was based on the OCM, and the procedures were relatively well-defined steps to be followed by a display-designer/model-practitioner. Zacharias and Levison[2] proposed an extension of this approach, for application to displays dominated by linear perspective cues. Again, the approach was model-based, using a hybrid of the OCM with a newly developed perspective cueing model. More recently, Garg and Schmidt[3] used the OCM to evaluate the effects of display dynamics on pilot performance and workload. Their model-based predictions showed excellent agreement with experimental results, validating the approach as a display analysis tool. The work reported here carries on this tradition of systematic model-based design, and extends the scope of applicability to the terrain-following cueing environment.

Experiment Description and Results

To provide an experimental basis for the validation of model-based display design, and to serve as a tool for developing and testing displays, a terrain-following simulation facility was developed and implemented. Figure 1 presents the overall block diagram of the terrain-following simulation, showing the basic modules and their interconnections. The terrain model, which drives the simulation, provides a terrain profile (h_t) having a power spectrum which matches measured profile spectra obtained from simulated TF missions[4]. Actual implementation is via generation of a sum-of-sines (SOS) signal, whose frequencies and amplitudes are chosen to match the desired spectrum, and whose phases are chosen to assure a random-appearing profile. The profile then drives the terrain-following guidance, which generates the desired flight path (h_g) to be followed by the pilot. The guidance system is implemented as a pure predictor and low-order filter approximation to the actual system being flown. The pilot views both the terrain profile (TP) and the desired flight path (DFP) on the terrain-following display, and generates a longitudinal control command, q_c, driving the pitch stability augmentation system (SAS), and, in turn, the aircraft longitudinal dynamics (elevator actuator dynamics are neglected in this simulation). The SAS is a simplified model of the actual aircraft SAS, since it ignores high-frequency structural/sensor compensation blocks. The dynamics are chosen to represent those of a strategic bomber flying at Mach 0.85 at sea level. The resulting aircraft states are then presented, directly and indirectly, on the TF display. Also shown in figure 1 is a lateral control loop which was included in the simulation to demonstrate more comprehensive flight scenarios, with occasional random heading changes between fixed course segments. The open loop lateral guidance system generates a heading command ψ_g, which the pilot cues on to generate a roll rate command p_c, which, in turn, drives a simplified model of the lateral dynamics. The resulting lateral aircraft states are also reflected in the TF display.

* Scientist, Member AIAA
** Scientist, Member AIAA
*** Principal Scientist, Member AIAA

The simulation is implemented on a Silicon Graphics Iris 3115, which is a 68020-based graphics computer with a Unix operating system. The pilot commands roll and pitch via a Measurement Systems Lab force stick, which is then filtered and sampled by a two-channel A/D converter. The pilot observes the real-time graphics on a 19" high resolution monitor.

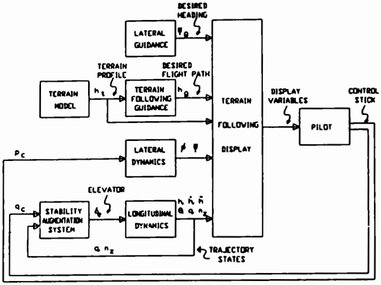


Figure 1: Overall Block Diagram for Terrain Following Simulation

The terrain-following display is designed to provide the pilot with the information needed to perform the TF task, without looking outside the cockpit. Two baseline display formats were studied: the vertical situation display (VSD) and a prototype pictorial guidance display (PGD). The results presented here derive from experiments performed with the VSD, enhanced versions of the VSD, and the PGD.

The simulated VSD, shown in figure 2, has two primary elements: 1) a forward-view artificial horizon; and 2) a side-view preview display. The artificial horizon shows sky ground separation, and includes pitch bars and an aircraft symbol. The preview display shows the terrain profile (TP) and the desired flight path (DFP), 60 seconds into the future. A crosshair representing the current aircraft position appears on the left edge of the preview display, and stays fixed in the display frame of reference; thus, the TP and DFP slide leftward and vertically as the vehicle flies over the terrain and changes altitude. Other elements of the VSD include an aircraft deviation pointer (ADP) to the left of the display, a high resolution bar display which indicates the current aircraft altitude error h_e; a digital radar altimeter; and a heading indicator with a desired heading pointer. The simulated VSD also supports a number of options and display enhancements. These include variable preview lengths for the TP and DFP; a gamma-track (GT) indicator in which two vectors show the aircraft's flight path angle, γ, and the current slope of the DFP under the aircraft, γ_{DFP}, both superimposed on the preview display; a flight director (FD) which computes and displays a desired pitch attitude based on current altitude error and the slope of the DFP; and a predictor (PR) indicator which shows the predicted aircraft altitude some t_p seconds ahead of the current aircraft position (patterned after the design proposed by Grunwald[5]).

The PGD, shown in figure 3, is a prototype display that provides a forward-looking perspective view of the TP and DFP superimposed on the artificial horizon display. Superimposed on the DFP itself is a "tunnel-in-the-sky" display denoted by its corners. This integrated pictorial view, given in proper perspective, replaces the side-view preview display elements found in the VSD. Other elements of the VSD incorporated in the PGD include the pitch indicator, the ADP, the radar altimeter, and the heading indicator.

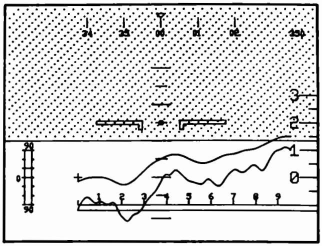


Figure 2: The Vertical Situation Display

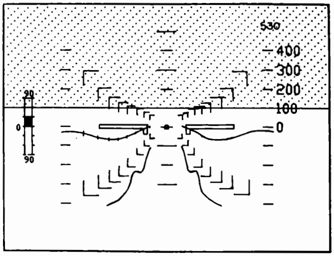


Figure 3: The Pictorial Guidance Display

The VSD (with and without enhancements) and the PGD were evaluated via simulation experiments, in terms of their ability to provide critical flight control information to the pilot. The basic approach involved: conduct of the simulation; real-time recording of the important time histories; time- and frequency-domain analysis of the histories; ensemble-averaging of these results across the subject population; and subsequent model-based analysis of the ensemble-averaged data. For the data presented here, ensemble averaging was conducted by first averaging across (typically 8) replications for a single subject and condition, to obtain single-subject means, and then averaging these means across subjects, to obtain across-subject statistics, for presentation and analysis.

An experimental series consisted of running a group of 4 or 5 subjects on a single display configuration, where each subject ran through a sufficient number of trials to cover training and final data collection. Three major series were conducted: one for each of four preview times (0, 4, 8, and 60 seconds) using the VSD without enhancements; a second for each of the 3 display enhancements using the VSD with the nominal preview time of 60 seconds; and a third using the prototype PGD. Before the experiments started, the subjects were briefed on the project and the TF task was thoroughly described to them. They were instructed to minimize the aircraft's altitude deviation from the DFP, with no conditions on any other states or display variables.

Figure 4 shows the effect of variable preview time. It graphically presents the population statistics of the performance or RMS error scores for the range of preview times studied in the first experimental series. Plotted for each of the four preview times are the across-subject means (solid circle) and standard deviations (error bar). It shows that the three preview conditions yield improved tracking performance and reduced stick activity, when compared with that obtained under no preview (zero seconds). Likely reasons for improved performance with preview include the ability to more accurately estimate aircraft and guidance states, the reduction of over-control or pilot-induced oscillations, and the ability to minimize altitude errors over some finite-length preview distance. There is little difference between the 4, 8, and 60 second preview cases except for the slight decrease in stick activity with increased preview. This indicates that most of the preview information that contributes to the pilot's control is contained within the first few seconds of preview. This confirms qualitatively similar findings by Tomizuka and Whitney[6] and Sheridan et al[7], where they showed decreasing effectiveness of preview beyond about 1 second, when working with more responsive vehicles and higher bandwidth driving inputs. For the TF preview display, these results imply that the DFP may not need to be computed beyond four or five seconds, if only immediate flight control precision is of consideration; additional preview may be necessary, however, to subserve more "outer-loop" functions, such as path guidance and terrain awareness.

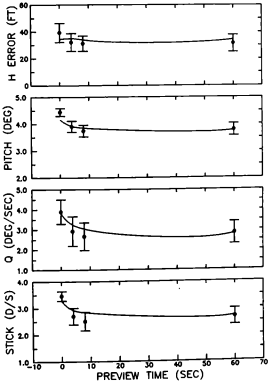


Figure 4: Effects of Preview on Population Scores

Figure 5 shows the across-subject ensemble statistics of the complex stick spectrum, broken into correlated (gain, phase), and uncorrelated (remnant) components, for the 60 second preview case. The correlated portion of the spectrum is computed from the normalized cross power spectral density (PSD) function, between the pilot's stick response u and an "effective" white noise w, via:

$$H_{wu}(\omega) = \Phi_{wu}(\omega)\,\Phi_{ww}^{-1}(\omega)$$

where Φ_{wu} is the noise-to-response cross PSD and Φ_{ww} is the noise PSD[8]. The noise itself is that which, when passed through an appropriate shaping filter, would yield the simulated terrain profile approximated by the simulator's sum-of-sines generator. The stick gain and phase are then obtained from

$$g(\omega) = \left|H_{wu}(\omega)\right|_{\omega=\omega^*} \quad ; \quad \phi(\omega) = \measuredangle H_{wu}(\omega)\Big|_{\omega=\omega^*}$$

where the ω^* are the sum-of-sines frequencies. The remnant or uncorrelated portion of the spectrum is calculated in the conventional manner, via

$$r(\omega) = \Phi_{uu}(\omega)\Big|_{\omega\neq\omega^*}$$

with a continuity approximation used to infer the $r(\omega^*)$ shown in the spectral plots.

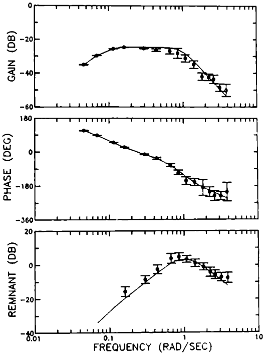


Figure 5: Pilot Frequency Response for the Vertical Situation Display

A number of characteristics may be noted by inspecting figure 5. First, the gain is fairly constant up to the break frequency of approximately 1 rad/s, indicating the bandwidth of effective pilot control. Second, the phase decreases at a relatively constant slope of 180 degrees per decade. Finally, the stick remnant peaks around 1 rad/s, corresponding to the break frequency of the gain.

Figure 6 presents performance scores for the nominal VSD (NOM), the enhanced VSD (GT, FD, PR), and the PGD. The following results are apparent when comparing the enhanced VSD and PGD to the nominal VSD. For the GT display the altitude score decreased while stick score remained approximately the same: better performance was achieved with the same amount of stick activity. We surmise that the GT enhancement reduced altitude error by displaying γ and γ_{DFP} in a coordinated format, enabling the subjects to accurately determine γ_{error}, an inner loop variable. The available phase lead from this cue then allowed them to increase their control effectiveness at higher frequencies. For the FD display the altitude error decreased slightly and the stick activity increased slightly. We would have expected more improvement in performance since the multi-cue VSD was replaced by a simple single degree-of-freedom target tracking display. The fact that the FD gains were not optimized (because of the study's limited scope), may have been the cause of this less than optimum performance. The PR enhancement produced the best performance as shown by the low altitude error and stick activity scores. The PR is most effective at the low to mid frequencies where it can accurately predict the effect of the pilot's stick input on the aircraft altitude, allowing the pilot to quickly observe and correct inputs which would contribute to altitude error. The PGD also produced much better performance than the nominal VSD. The tracking error and stick activity have been reduced to levels comparable to that obtained with the PR. This was accomplished by using the exact same display elements found in the nominal VSD. The difference is that the PGD presents the elements in a coordinated and more natural format. The PGD format is also egocentric so it matches the pilot's cognitive model of the outside world, thus making his control response more intuitive.

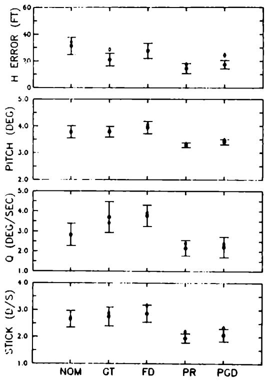


Figure 6: **Effects of Display Enhancements on Population Scores**

Model-Based Analysis of Simulator Data

Model-based analysis of the ensemble average simulation data for various display configurations provides for the transformation of this data into a more compact set of model parameters. These model parameters provide insights into the interpretation of the experimental results as well as the potential for extrapolation beyond the experimental data set. The approach centers on the use of an integrated pilot/vehicle/display model as shown in figure 7. This model relates pertinent vehicle and display system characteristics to relevant human visual perception processing and flight control strategies. It consists of two component elements: the Optimal Control Model (OCM) models the pilot's information processing and continuous control activities; and a Visual Cueing Model (VCM), comprised of four perceptual sub-models, is used to model the pilot's interaction with his display environment and his resulting perceptual performance. We further briefly describe the OCM and the sub-models of the VCM before presenting the model-based analysis results.

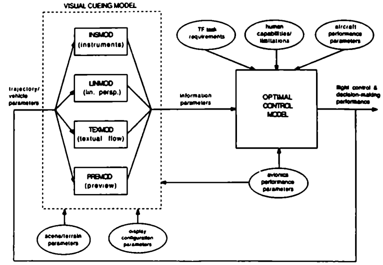

VISUAL CUEING MODEL
INSMOD (instruments)
LINMOD (lin. persp.)
TEXMOD (textual flow)
PREMOD (preview)
trajectory/ vehicle parameters
information parameters
TF task requirements
human capabilities/ limitations
aircraft performance parameters
OPTIMAL CONTROL MODEL
flight control & decision-making performance
avionics performance parameters
scene/terrain parameters
display configuration parameters

Figure 7: Integrated Pilot/Vehicle/Display Model

The OCM has been developed within the systems framework of modern estimation and control theory[9]. The basic assumption underlying the model is that the well-trained well-motivated human operator behaves optimally in some sense, subject to inherent psychophysical limitations which constrain the range of his behavior. In the flight control environment, the model is capable of predicting steady-state task performance, frequency-domain pilot transfer functions, and frequency-domain pilot remnant. A general block diagram of the OCM is given in figure 8. The system portion (outside the dashed box) provides for representations of control interface dynamics and system (vehicle) dynamics. As shown, the two inputs to the system are the set of controls generated by the operator ($\underline{u}$), and the system disturbances which act to drive the overall loop. The set of system outputs processed through the display interface is a multi-modality cue set driving the pilot's various sensory systems. Limitations in the pilot's ability to process information displayed to him are accounted for by translating the observed variables $\underline{y}$ into a set of delayed, noisy, perceived variables $\underline{y}_p$. The optimal estimator, predictor, and controller represent the adjustments or adaptations made by the pilot to optimize his behavior under these non-ideal conditions, and to account for the dynamic interaction of the system he is controlling. The neuromotor dynamics and motor noise account for bandwidth limitations of the human, and his inability to generate noise-free controls.

Three of the visual cueing submodels shown in figure 7 are used for modeling the various

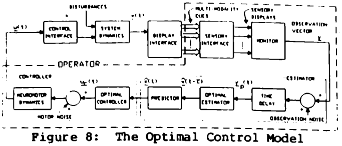


Figure 8: The Optimal Control Model

terrain-following display configurations: an instrument cueing model (INSMOD), a textural cueing model (TEXMOD), and a preview cueing model (PREMOD). INSMOD, a direct submodel of the OCM, models the operator's processing of visual cues presented via conventional instrument displays such as a pointer/bar display (e.g., the VSD's ADP), or a flight director display. The model assumes that for simple instruments the pilot sees the display variable and its time rate-of-change. To use the model, the variable being displayed is specified, as well as display factors such as: resolution limitations, zero reference uncertainties, and acuity thresholds. These are then transformed into associated observation noise levels for direct use in the OCM. The submodel TEXMOD has application in the analysis and modeling of scenes dominated by textural cues. The model is predicated on the notion that the pilot makes noisy, sampled measurements on the spatially-distributed optic flow-field surrounding him, and on the basis of these measurements, generates estimates of his own linear and angular velocities. The accuracies of these estimates then feed the OCM display and observation noise model. Finally, the modeling of previewed path information, such as the desired flight paths and terrain profiles of the VSD and PGD, is accomplished via the use of the submodel PREMOD[10]. The approach involves fitting a parametric curve/surface model to the preview path, to generate estimates of the path's slope, curvature and higher derivatives. In the present study, a second-order polynomial fit was used for the previewed DFP, and a first-order fit for the previewed terrain profile. The resulting augmented display vector, and the associated display thresholds, were then fed to the OCM observation noise model, for further processing. Additional details on all of the above visual cueing models are to be found in Zacharias[11].

These pilot/vehicle and visual perception submodels were used to analyze the performance scores and frequency-domain data for the various display configurations of the terrain-following simulation. We specified the display variable set, attention allocation, and visual cue thresholds for each display configuration. Model-based analysis was then conducted to identify a set of model parameters that provided a best match between model-generated data and simulator data. For this study, simulation data matching was accomplished via a model parameter identification scheme as proposed by Jain[8]. The approach provides for a direct match of the pilot's complex stick spectra, as well as remnant spectra and performance scores.

We conducted model-based analysis of the simulations using the VSD with variable preview (0, 4, 8, and 60 seconds). We found that suitable fits could be obtained with the simple display model assumptions noted above, but that improved matching of the data trends could be obtained if it was further assumed that, with different preview intervals, the subject pilots: a) reallocated their attention-sharing to maximize performance; and b) readjusted their control objective (weights) to de-emphasize short-term tracking when faced with long-term preview intervals.

These assumptions are shown in Table 1, showing the model parameters obtained. The table gives the base pilot parameters, the attention allocation among the various display elements, and the control weights. The base pilot parameters are: 1) a motor noise ratio that accounts for random errors in intended control and the fact that the pilot may not have perfect knowledge of his own control activity; 2) an observation noise ratio that accounts for errors in observing displayed variables; 3) a motor time constant associated with the pilot's neuro-motor dynamics; and 4) a perceptual time delay associated with the lumped effects of visual, central, and motor processing pathways. These base parameters remain fixed over the various preview times and assume values that are typical of this type of low bandwith control task. This includes the motor time constant of .44 sec, larger than most values encountered in laboratory tracking tasks but not totally unexpected given the system's bandwidth limitations. The attention allocation parameters reflect the changes in attention with preview time, needed to optimize task performance. Note the reduction of attention on the pitch and DFP cues with the presence of preview. Finally, the cost parameters reflect the pilot's de-emphasis of short-term tracking performance: as preview increases, more control emphasis is placed on minimizing vertical acceleration and less on minimizing instantaneous vertical path errors and rates.

Table 1: Model Parameters for VSD with Variable Preview

	NO PREVIEW	4 SECOND	8 SECOND	60 SECOND
Base Parameters				
Motor Noise Ratio (dB)	-38.6	-38.6	-38.6	-38.6
Observation Noise Ratio (dB)	-17.1	-17.1	-17.1	-17.1
Motor Time Constant (s)	0.44	0.44	0.44	0.44
Perceptual Time Delay (s)	0.25	0.25	0.25	0.25
Attention Allocation				
pitch	30%	10%	10%	10%
DFP	30%	10%	10%	10%
ADP	40%	70%	70%	70%
previewed DFP		10%	10%	10%
Cost Weightings				
guidance error (ft)	1	1	1	1
guidance error rate (ft)		1	1.3	1.3
vertical accel. (ft/sec2)		4.6	1.7	1.2
pitch rate (deg/sec)		1	1.4	1.5

The resulting matches to the performance scores are shown as the smooth curve given earlier in figure 4. The model confirms the data implications concerning the importance of preview for short preview intervals for the first 5 seconds or so. This same modeling effort also accounts for the observed pilot frequency response trends, illustrated by the curve given earlier in figure 5 for the nominal VSD with 60 second preview. Clearly, the model curves provide a close match to the across-subject empirical data means, across the full frequency bandwidth of interest. Similar quality matches were obtained for the other preview intervals studied.

Modeling of the three VSD enhancements is accomplished via augmentation of the nominal VSD with the necessary additional display elements associated with the particular enhancement. For example, the GT enhancement provides derivative information on the terrain profile and desired flight path; accordingly, we augmented the nominal display vector by adding the variables $\dot{h}$, $\dot{h}_{DFP}$, and $\ddot{h}_{DFP}$. The thresholds associated with these augmented variables are then calculated from either existing VSD elements or actual display resolution. A similar procedure is used to model the flight director and predictor.

The resulting model-based analysis showed that the effect of the VSD enhancements can be readily accounted for with a few additional assumptions, which are summarized in Table 2. For the GT enhancement, attention is shifted from the ADP to the GT symbology, and less control emphasis is placed on minimizing vertical acceleration, and more on path error and error rate. With the FD enhancement, full attention is on the flight director itself and the aircraft's pitch bars, and control emphasis is totally on path error. Finally, for the PR enhancement, the base parameters show a reduction in observation noise, while attention is shifted from the ADP to the PR symbology and the previewed DFP. Also, greater control emphasis is placed on the path error. These relatively straightforward changes in the pilot's attention-allocation strategy and control objectives, along with slight variations in some of the base parameters, provide us with a direct means of accounting for changes in the pilot's control behavior and task performance with each enhancement. The resulting model matches to performance scores are shown in figure 6 for the nominal VSD and its enhancements. The simulation data is seen to be well-matched by the model points indicated by the diamonds. Corresponding matches to the frequency-domain data are equally close.

Table 2: Model Parameters for VSD with Enhancements

	BASELINE	GT	FD	PR
Base Parameters				
Motor Noise Ratio (dB)	-38.6	-38.6	-38.8	-40
Observation Noise Ratio (dB)	-17.1	-17.4	-17.1	-20
Motor Time Constant (s)	0.44	0.4	0.44	0.4
Perceptual Time Delay (s)	0.25	0.23	0.25	0.23
Attention Allocation				
pitch	10%	5%	100%	5%
DFP	10%	5%		5%
ADP	70%	35%		30%
previewed DFP	10%	20%		30%
gamma-track		35%		
flight director			100%	
predictor				30%
Cost Weightings				
guidance error (ft)	1	1	1	1
guidance error rate (ft)	1.3	1.2		6.4
vertical accel. (ft/sec2)	1.2	8.8		2.9
pitch rate (deg/sec)	1.5	3		8

Modeling the pilot's use of the PGD requires that we recognize the presence of a set of dynamic image-flow cues, in this case the corner elements of the PGD tunnel. In accordance with the visual submodel TEXMOD, these cues support the estimation of vehicle aimpoint and surface-normal impact time (that is, altitude in flight time seconds above the terrain). Errors in estimating longitudinal plane aimpoint correspond to errors in inferred flight path angle, γ. Errors in estimating impact time correspond to errors in inferred altitude error, h_e. To generate these estimation errors, Monte Carlo simulations were run using TEXMOD, with the parameters specifying the visual environment (e.g., vehicle speed, geometry of the tunnel corners, etc.) chosen to simulate the PGD display situation. Noisy observation of the tunnel corners were simulated by additive measurement noise corrupting the flow-field measurements. Ensemble statistics of the resulting estimation errors were then computed to generate threshold values for the two inferred cues (γ, h_e), which were then fed to the OCM to generate predictions of closed-loop performance and used as the desired flow-field cue threshold specifications.

The PGD modeling effort also required additional assumptions regarding attention allocation and control emphasis. Specifically, the analysis showed that attention needs to be split between the ADP and the integrated tunnel display, and more control emphasis needs to be placed on tracking error and error rate. These trends are summarized in Table 3. The resulting model match for the performance scores is shown in figure 6. Again, the simulation data is seen to be relatively well-matched by the model. The same is true for the frequency-domain data.

Table 3: Model Parameters for PGD

	VSD	PGD
Base Parameters		
Motor Noise Ratio (dB)	-38.6	-38.6
Observation Noise Ratio (dB)	-17.1	-17.4
Motor Time Constant (s)	0.44	0.44
Perceptual Time Delay (s)	0.25	0.23
Attention Allocation		
pitch	10%	
DFP	10%	
ADP	70%	40%
previewed DFP	10%	
integrated display		60%
Cost Weightings		
guidance error (ft)	1	1
guidance error rate (ft)	1.3	0.9
vertical accel. (ft/sec2)	1.2	1.9
pitch rate (deg/sec)	1.5	0.9

Model-Based Display Design Predictions

With model analysis of these display configurations in hand, it is possible to then use the model to predict the effects of varying certain display elements in order to optimize them. For this effort, we considered two specific display configurations: the VSD with the flight director (FD), and the VSD with the predictor (PR). With the first, we varied the FD gain relating path error h_e to FD symbol displacement; with the second, we varied the PR prediction time. In both cases, we attempted to determine the optimum choices in terms of predicted closed-loop tracking error. Model-based results were identified using the baseline pilot parameters generated earlier for the two enhancement options. Comparison of pre-simulation model predictions with subsequent simulation results are shown in figure 9 for the flight director (for two individual subjects), and figure 10 for the predictor (for an ensemble of 3 subjects). Model predictions are shown by the curves, while data means and standard deviations are presented by solid circles and error bars. The flight director results of figure 9 confirm the relative insensitivity to FD gain predicted by the model, and identifies a relatively shallow optimum in the vicinity of 0.10. The predictor results of figure 10 also show a reasonable match to the data trends, with the optimum prediction time occurring at about 5 seconds, where both altitude error and stick activity are low. The disagreement between model and data at the 4-second prediction time may be a result of (extended) training effects since it was the first set to be run with the predictor.

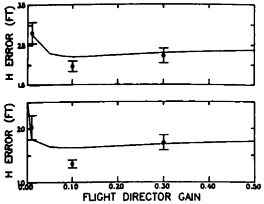


Figure 9: Flight Director Law Design - Model-Based Procedure (Two Subjects)

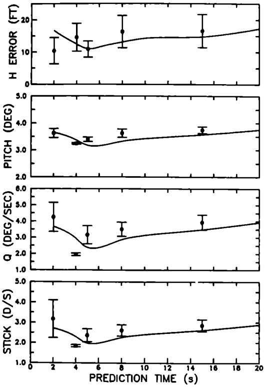


Figure 10: Predictor Law Design - Model-Based Procedure

The analysis and design of the flight director and predictor laws serve as prototypes of a general model-based display design method. In its most summary form, this method is comprised of the following basic steps.

Step 1: Define Non-Display Systems-Related Task Parameters

This would include a specification of vehicle characteristics, (such as the vehicle dynamics and SAS parameters), environmental features (such as terrain shape and gust disturbance level), and task requirements (such as desired path following precision). The equations of motion are written in standard state-space format, to support an input-output system description from the point of view of the pilot.

Step 2: Define Display-Related Parameters

This includes a specification of basic display type (instrument, perspective, textural), and corresponding display attributes (such as resolution, field-of-view, etc.).

As described earlier, each element of the display presents information about a physical variable which combines the system states in some unique way. The display designer must consider the candidate display, and with the help of the VCM sub-models, define the unique combination of states which describes each display variable. Once the system outputs are defined (i.e., the physical variables displayed) and related to the system states, the display presentation accuracy must be specified. This is implemented via threshold values, which are associated with either limitations in the system display, or with human perceptual limitations.

Step 3: Define Overall System

With the display equations defined, the designer has all the system elements described, and it remains for him to describe how the subsystems interlink with each other, and how they interface with the pilot. Defining the system architecture, with the given display and non-display elements then specifies the overall system.

Step 4: Define Human-Related Model Parameters

These are associated with perception, information-processing, and decision/control. These would include standard "textbook" parameters (such as neuromotor lag time), as well as parameters that might be influenced by the task itself (such as fractional attention due to side-task workload). These also include the cost function weighting parameters which serve to define the pilot's objective function and his flight task.

Step 5: Exercise Overall Model

Simulate expected task performance with the given display configuration. Predict performance measures appropriate for display evaluation, such as attentional allocation, perceptual accuracy, flight control performance etc. Conduct design trade studies, to investigate the impact of variations in display configuration on performance.

Step 6: Iterate On Basic Display Design

Repeat steps 2 through 5, as required, until a suitable display is arrived at. Validate the results via man-in-the-loop simulation, over a small select group of candidate display designs.

We believe that this systematic approach, with its reliance on a rational documented model of pilot/vehicle/display performance, can minimize the subjectiveness required in current display design efforts. It can take full advantage of both our knowledge of current and anticipated display technology, and of our understanding of the human pilot's ability to acquire, assimilate, and act on displayed information, and it allows us to integrate both sources of knowledge in the display design process.

Summary

We have outlined and demonstrated the use of a model-based method to evaluate and design terrain-following cockpit displays. Two basic display configurations were studied: the nominal vertical situation display (VSD) along with several enhancements, and a pictorial guidance display (PGD). A real-time terrain-following simulation facility was developed and exercised to generate an experimental data base. Model-based analysis demonstrated the ability to closely match performance and frequency response data across the range of display configurations studied, accounting for both general performance trends and fine-grained pilot dynamic response strategy in the measured data. Model-based optimization of flight director and predictor laws demonstrated how the method can be extended beyond display evaluation, to directly support pre-simulation display design and optimization.

Acknowledgments

The work reported on here was performed under sponsorship by the Armstrong Aerospace Medical Research Laboratory (AAMRL), under USAF/ASD Contract F33615-86-C-0551. The authors thank the Technical Monitors, Mr. Brad Purvis and Lt. Ralph St. John of AAMRL, for their continued support and encouragement on this project. We also thank Ms. Pauline O'Donnell of Charles River Analytics Inc. for her skillful creation and editing of this paper.

References

[1] Hess R.A., "Analytical Display Design for Flight Tasks Conducted Under Instrument Meteorological Conditions," IEEE Trans. on Systems, Man, and Cybernetics, SMC-7, 1977.

[2] Zacharias, G.L., and Levison, W.H., "A Model-Based Procedure for Determining Visual Cue Requirements," Proceedings of Image Generation/Display Conf. II, Williams Air Force Base, AZ, June 1981.

[3] Garg, S. and Schmidt D.K., "Model-Based Evaluation of Display-Dynamics Effects in Pursuit Tracking," Proceedings of the 22nd Annual Conference on Manual Control, Air Force Technical Report AFWAL-TR-86-3093, July 1986.

[4] Brinkley, C.W., Sharp, P.S., and Abrams., R., "B-1 Terrain-Following Development," AGARD Conf. Proceedings, 1977.

[5] Grunwald, A.J., "Predictor Laws for Pictorial Flight Displays," Journal of Guidance, Control, and Dynamics, Vol. 8, No. 5, Sept.-Oct. 1985.

[6] Tomizuka, M., and Whitney, D.E., "The Preview Control Problem with Application to Man Machine System Analysis," Proc. of the Ninth Annual Conference on Manual Control, MIT, Cambridge, MA, May 1973.

[7] Sheridan, T.B., Merel, M.H., and Kreifeldt, J.G., "Some Predictive Characteristics of the Human Controller," presented at the AIAA Guidance and Control Conference, Cambridge, MA, August 1963.

[8] Jain, A., "Modified Parameter Identification Scheme for the OCM," Technical Memo TM8804, Charles River Analytics Inc., Cambridge, MA, March 1988.

[9] Kleinman, D.L., Baron, S., and Levison, W.H., "A Control Theoretic Approach to Manned-Vehicle Systems Analysis," IEEE Trans. on Auto. Control, AC-16, December 1971.

[10] Brun, H.M., and Zacharias, G.L., "Model-Based Methodology for Terrain-Following Display Design," Report No. R8603, Charles River Analytics Inc., Cambridge MA, February 1986.

[11] Zacharias, G.L., "An Estimation/Control Model of Egomotion," in Warren, R. (ed.), Perception and Control of Self-Motion, Lawrence Erlbaum Assoc. Inc., Hillsdale, NJ, 1989.

89-3292-CP

HELMET MOUNTED DISPLAY APPLICATIONS FOR ENHANCED PILOT AWARENESS

Gregory M. Hardyman
Michael H. Smith

Lockheed Aeronautical Systems Company
Burbank, California

Abstract

In today's highly sophisticated fighter cockpits with their advanced sensor capabilities, there is a substantial amount of complex data presented to the fighter pilot. The filtering and presentation of this data to provide the pilot with high quality situational awareness provides one of the primary challenges for effective crew systems design. The Helmet Mounted Display promises to serve as an excellent tool for presenting some of this data and providing the pilot with another method to interact with his aircraft.

One of the most significant advantages of using the HMD involves providing the pilot with effective situational awareness information while he maintains his attention on the external heads up environment while minimizing heads down display examination. The helmet can also be used effectively as a visually directed selection device that can save precious seconds in time critical pilot decision tasks. Potential uses for an HMD include target location queuing to enhance visual acquisition; heads up flight and avionics data; tactical, evasive, and pilot response ques; target designation for sensor prioritization; weapon assignment and cooperative engagement coordination; and heads down cursor slewing to reduce heads down interface time.

This paper outlines the basic HMD integration design at Lockheed's Weapon System Simulation Center and some of the basic issues involved with helmet mounted sights. These include tracking accuracy errors caused by geometric, operational, and human variance considerations. This paper will also discuss possible HMD application areas which appear promising to reduce pilot workload. The discussion will include benefits, potential problems, and workable solutions for many application areas. Some human factors studies will be suggested which would help evaluate the effectiveness of the HMD for these application areas.

The Weapon Systems Simulation Center at Rye Canyon supports these types of crew system design studies in a real-time full mission visual flight simulation. The WSSC external battle environment provides a high fidelity analysis capability which can provide very meaningful evaluations that can directly influence fighter aircraft system designs of the future.

Introduction

Background Discussion

The amount of information available to today's fighter pilot is overwhelming. The task of filtering and presenting this data to provide high quality situational awareness is a fundamental challenge of good crew systems design. Ergonomic and human factors studies are continually evaluating new Pilot Vehicle Interface technologies and display formats. Many new technologies such as MFD's, touch screens, voice warning, voice recognition, and Helmet Displays are commonly evaluated for use in today's fighter aircraft. This paper intends to outline the basic issues involved in installing and evaluating the use of a Helmet Mounted Display system at Lockheed's new Weapon Systems Simulation Center at Rye Canyon, California.

Pilot workload is of critical importance in today's battle environment. As new high technology weapon systems are developed, the battle is shifting from shorter range missile and gun combat situations to Beyond Visual Range engagements. In response to this trend, PVI designs are using Multi-Function Displays to provide the pilot with graphical representations of his mission objectives, weapon stores, operational status, and of the external threat environment. The thrust of these designs is to keep the information and display designs as simple as possible while still providing the pilot with the essential information required to defeat his enemy and survive. One problem with these displays is that they require a pilot to focus his attention inside the cockpit. While you might think that this should not be a significant problem in BVR engagements, experience has proven otherwise. Pilots who have their heads down can miss many basic visual observations such as; missile launch detections, undetected enemy aircraft, and ground collision. The HMD offers the potential of providing the pilot with essential situational awareness while maintaining his attention out the window.

The WSSC HMD Environment

A typical Helmet Mounted Display consists of a pilot's helmet, some type of tracking system to identify head movements, and a graphics display system to provide information.

WSSC's HMD simulation hardware consists of a Gould 97/80, A Kaiser helmet mounted sight, a Kaiser display processor, a Polhemus electronic processing unit, an Evans & Sutherland PS-350 stroke graphics engine, a VME 80386 microprocessor, and a 1553 bus controller card. the Gould 97/80 is one of 9 in the WSSC real time simulation environment and is dedicated for avionics processing. Pilot head movements are detected and recorded by the Polhemus magnetic tracker. These head movements are converted into x, y,z translations and yaw, pitch, roll values by the Polhemus electronic processing unit. Data is then transmitted to the Gould via a 1553 bus at 60hz. HMD driver software on the Gould reads this data out of predefined memory locations at 30hz and uses the HMD tracking information in avionics and graphics computations. The Graphics driver then buffers display commands for the PS-350 that are shipped via a Gould HSD to a VME 80386 microprocessor that filters the data, converts the floating point to IEEE standard and issues op codes to the PS-350 graphics engine. Stroke graphics are generated at 30hz and the output video signal is then shipped to the Helmet Mounted Display for viewing.

HMD Uses and Issues

The Helmet Mounted Display has been proposed for a variety of uses. These include; heads up information, pilot response and warning cues, heads up sensor sluing and designations, and heads down cursor sluing and designations.

Discussion

Real Time Implementation

The Kaiser HMD System

The Kaiser helmet mounted sight used at WSSC consists of a 48 ounce helmet with special insertable liners that are heated in a microwave oven before being form fitted for an individual pilot. The helmet visor is a clear plastic mask with a 12 degree fov optical inset aligned parallel with the pilots right eye. An internal mirror is used to reflect the graphics image onto the 12 degree inset. There is also an external adjustment lever on the helmet that allows the pilot to properly collimate the graphics image onto a G.E. Compu-Scene IV dome surface 14 feet away.

There is a Polhemus magnetic sensor that is installed in the upper center rear of the helmet. This sensor is used to track helmet movements in relation to a three point magnetic reference device that is installed in a fixed position above the pilots head. Any substantial metal structures in the cockpit can have an adverse affect on tracking accuracy. One of the initial installation requirements of installing a Polhemus tracker is to magnetically map the cockpit environment and load this data into the Polhemus electronic processing unit.

Tracker and video data is shipped via a cable to a Kaiser display processor. This box handles both the PS-350 graphics input signals, and routes the HMD tracking data to the Polhemus electronic processing unit. The EPU is the brain of the tracking system. Output data from the helmet sensor is then filtered through the magnetic map and converted into an x,y,z translations and yaw, pitch, roll head movements. The data is then shipped to the Gould 97/80 via a 1553 bus.

Dome and Cockpit Geometry

The cockpit and dome geometry precision that is required by the HMD is critical. First, the G.E. Compu-Scene IV computer image generator uses the center of the dome as the visual eye point. The cockpit design eye must also use this dome center reference to avoid perspective, lighting, and dome distortion errors. The center of gravity, i.e., the rotation focal point of the aircraft, is then defined in the CIG as an offset from viewpoint. The helmet boresight reference system of the helmet after boresighting should align parallel with the cockpit axis. Only after all of these preconditions are satisfied can accurate helmet sighting and tracking measurements be taken.

In the WSSC dome #1, these geometry issues have not to date been fully implemented and verified. The pilot design eye point is 8-10 inches off of dome center. This alone will cause a 2 degree parallax effort at 14 feet. Helmet collimation can only focus the graphics image at 14 feet, but has no affect on eye point parallax. This kind of error is greatly reduced in a true aircraft environment. A two degree parallax error at 14 feet would become a .02 degree error at 1/4 mile.

Graphics Display Geometry

Graphics displays for the HMD must be concerned with simplicity, clarity, and geometric precision. Display geometry should first compute a correct offset reference from the aircraft CG. The aircraft rotation matrix should then be applied to this offset to compute a true earth reference offset which is them added to the aircraft position to determine a precise viewpoint location. The rotation matrix used by the helmet is a standard 3-D rotation matrix using (+yaw, -pitch, -roll).

Display simplicity and clarity are also very critical to making the helmet a useful tool. The pilots intent for using a helmet is to maintain his attention on events out his window. Flashing symbology and display clutter are distracting to a pilot and should be avoided in most cases. The HMD uses stroke graphics to provide straight lines and clean sharp image resolution. WSSC uses a 30hz update to sync with power supplies, simulation data rates, and provide very smooth graphics movement.

Update Rates, Transport Lags, and Data Synchronization

Update rates are very important in HMD displays. Slow update rates are guaranteed to induce displays that step and lag pilot head movements. If the update rates are too fast, they impose substantial computational and data bandwidth problems with no significant performance gain. At WSSC we use a 30hz HMD update rate for the following reasons; 30hz is in sync with US electric power frequencies and is thus commonly used by nearly all graphics devices. A 30hz rate is the minimum synchronized data rate that appears as fluid motion to the human observer, and 30 hz is in sync with the base 120hz rate used at the WSSC simulation.

Data synchronization is concerned with data arrival patterns. If the HMD display runs at 30hz but the data that updates the graphics is computed at 20hz, we have a synchronization problem. What really happens is that for every three times the HMD graphics reads new display values, only two are available. This would cause the visual effect of move, move, stop, move, move, stop. Which is perceived as a 10hz stepping effect. If the data is computed at 35hz, we would sample the data 30 times each second and five of those times we would get a double increment of data. This would cause a 5hz jumping effect. Also, this data would not have a consistent transport lag time alignment with the display. Dynamic time alignment corrections is what this stalling and jumping effect really is all about.

Even two processes that compute data at 30hz can display adverse data synchronization effects. Generally, 30hz data is generated each 1/30th of a second plus or minus a delta time that accounts for conditionally processed events. In concurrent processing systems, this factor may cause a data event that was computed just before a read in one frame, to be late and miss the read the next frame. This type of synchronization problem can cause very intermittent stops and jumps in graphics motion. The normal solution to this problem is usage of event flags and semaphores. In real time simulation, this type of overhead is just too costly. In real time simulation, well designed code can usually minimize this type of problem.

Transport lags pose another problem in simulation displays. Transport lag is the delay between an event and a response. If these lags get too great, the pilot is unable to correctly associate events and responses. If the dome visuals react too slowly to the pilot's stick input, the pilot will tend to over control the aircraft which may result in PIO effects. Even lags of 100ms are often considered too large to evaluate the handling qualities of an air vehicle. Significant lags in the HMD will cause sluggish response to pilot head movements. Also, if the dome image lags are much different then the helmet lags dynamic visual alignment, errors are perceived. This type of error is not generally a problem for most HMD users but would be significant in high dynamic situations such as with a HUD LCOS gun sight.

Tracking Accuracy Issues

Tracking accuracy is a critical measurement in determining the potential applications and usefulness of a HMD. It should be obvious by this point that there are many factors that affect tracking accuracy. If any of these aforementioned issues exist, then meaningful tracking accuracy measurements may be impossible to derive. In fact, testing HMD applications in a dynamic real time full mission dome simulation may be an unsuitable environment for meaningful crew systems analysis of some HMD applications.

Tracking errors are caused by natural pilot head jitter which usually result in a one to three degree yaw and pitch oscillation of a pilot's head while focusing on a fixed position. Recording and magnetic mapping errors are present in the Polhemus tracking system. These were usually only a fraction of a degree. Helmet boresight errors can cause large parallax and rotation axis misalignment. Anthropometric errors are caused by the parallax of right eye alignment variance through the helmet sight. Dome and cockpit geometry errors, graphics geometry errors, update rates, transport lags, and data synchronization errors add to the total error. WSSC experienced overall tracking errors of close to six degrees. In a 12 degree field of view HMD this can pose a real problem.

WSSC has recently received a new 20 degree FOV Kaiser helmet and has completed a second 28 foot dome which has taken into account most of the dome and cockpit geometry problems. We are currently measuring system transport lags, evaluating data synchronization issues. I expect that with careful planning, tracking errors could be reduced to the 1-2 degree range.

Real Time Applications

General HMD Applications

One of the primary tasks of a Helmet Mounted Display is to provide essential information to the pilot while his attention remains out the window. This data would generally not compete with information normally available on the Heads Up Display. But, the full panoramic viewing aspects of the HMD enable it to become an excellent directing device.

HMD systems are currently in use in attack air vehicles such as the AH-64 Apache helicopter. Pilots can slave sensor systems to their line of sight. They can also direct guns and other weapon systems. the HMD can also serve as a warning device that directs the pilots attention to an area of immediate concern. The Apache helicopter also uses the HMD to designate targets. the HMD has proven itself to be a very valuable tool on board this attack helicopter. It may also prove to be valuable on hovering attack aircraft such as the AV-8B Harrier. Maybe the HMD can serve a useful tool on all attack aircraft, but high speed and dynamic flight profiles define new problems for HMD use.

At WSSC, we are currently evaluating the potential use of a HMD on board the Advanced Tactical Fighter. the use of a helmet system for beyond visual range engagements is relatively new. The role of a helmet for BVR/WVR transitions is also under study. The role of the HMD in close in combat may be just the right technology to give out pilots an extra edge.

Heads Up Designations

At WSSC, heads up designations have been implemented on the HMD in a BVR environment. Initial pilot reaction was ambivalent since in a BVR engagement a heads down designate is adequate. There is not an significant advantage to designating heads up. The area where heads up designates have promise is in a BVR to WVR transition.

BVR/WVR Transitions

Pilot feedback has indicated that heads up designates in the transition made would be a good attribute. WSSC is currently in the process of generating this capability. The pilot's like the idea of slaving their head, designating, and beginning WVR tactics with the Helmet. Future studies at WSSC will provide meaningful data and information on this subject.

WVR Applications

All pilots at WSSC agreed on the usefulness Helmet designating and queuing would have in a WVR situation. Keeping eyes out and obtaining useful information on the helmet while doing so is of critical importance. When split second decisions are being made, the pilot does not have time to go heads down. WSSC is currently implementing a WVR capability using a helmet mounted display.

Conclusion

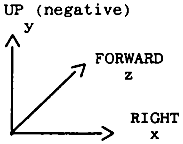
Flying with the helmet in a BVR Scenario, WSSC has demonstrated queuing and designating capabilities. The Helmet proved adequate in designating targets and allowed tracking of a target through sluing. Que arrows were helpful in directing a pilot where to search for a target no longer on the display. Further studies in a BVR/WVR transition or WVR environment will determine how valuable que arrows will be. A question mark remains as to how much error can be tolerated while using the helmet. Future studies will determine how the transport lag, geometry errors, target projector accuracy, and 14 feet dome radius induced accuracy errors, etc. can be minimized. If the minimization can reduce the error in target alignment of the Helmet vs dome then the HMD will be a valuable tool for scenario studies.

APPENDIX A

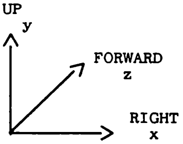
HMD ALGORITHMS

The method for placing targets on the helmet mounted display utilizes 3-D matrix transformations. In addition, target data is supplied relative to the ownership in which the helmet resides. Using relative data, the targets are run through an inverse matrix of helmet yaw pitch and roll. This effectively allows a solution for target field of view calculations as if helmet yaw pitch and roll were 0, 0, 0. The azimuth and elevation of a target are used to display it in x, y coordinates on a PS-350 graphics engine.

Relative target data provided by avionics is of the following format:



The PS-350 graphics engine is of the following format:



The following assignments are made:

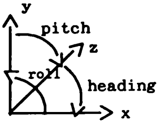
$$X_{350} = X_{AVI}$$

$$Y_{350} = -Y_{AVI}$$

$$Z_{350} = Z_{AVI}$$

Sign changes associated with the yaw pitch and roll of the helmet are required for proper orientation. Avionics data produces positive roll from the Y axis to the X axis. In the PS 350, it is from the X axis to the Y axis. Therefore, $\text{roll}_{350} = -\text{roll}_{AVI}$.

Avionics data produces positive pitch from the Z axis to the Y axis. In the PS-350 it is from the Y axis to the Z axis. Therefore, pitch 350 = $-\text{pitch}_{AVI}$.



PS 350 POSITIVE ROTATE DIRECTIONS

To run the helmet yaw, pitch, and roll data through an inverse matrix to reach an effective 0, 0, 0 roll, pitch, yaw point for the helmet requires negating the pitch roll yaw values first.

roll = -roll
yaw = -yaw
pitch = -pitch

At this point, the inverse matrix operation may be performed.

The inverse matrix operation is:

$$\begin{bmatrix}\text{yaw}\\ \text{matrix}\end{bmatrix} \quad \begin{bmatrix}\text{pitch}\\ \text{matrix}\end{bmatrix} \quad \begin{bmatrix}\text{roll}\\ \text{matrix}\end{bmatrix}$$

Substituting produces the following:

$$\begin{bmatrix}x'\\ y'\\ z'\end{bmatrix} =$$

$$\begin{bmatrix}x\\ z\\ y\end{bmatrix}\begin{bmatrix}\cos(yaw) & 0 & -\sin(yaw)\\ \sin(yaw) & 0 & \cos(yaw)\\ 0 & 1 & 0\end{bmatrix}\begin{bmatrix}1 & 0 & 0\\ 0 & -\sin(plt) & \cos(plt)\\ 0 & \cos(plt) & \sin(plt)\end{bmatrix}\begin{bmatrix}\cos(rll) & \sin(rll) & 0\\ 0 & 0 & 1\\ -\sin(rll) & \cos(rll) & 0\end{bmatrix}$$

This becomes:

$$\begin{aligned}
x' &= x{*}(\cos(yaw){*}\cos(rll)-\sin(yaw){*}\sin(plt){*}\sin(rll))\\
&\quad -y{*}(\cos(plt){*}\sin(rll))\\
&\quad +z{*}(\sin(yaw){*}\cos(rll)+\cos(yaw){*}\sin(plt){*}\sin(rll))\\
y' &= x{*}(\cos(yaw){*}\sin(rll)+\sin(yaw){*}\sin(plt){*}\sin(rll))\\
&\quad +y{*}(\cos(plt){*}\cos(rll))\\
&\quad +z{*}(\sin(yaw){*}\sin(rll)-\cos(yaw){*}\sin(plt){*}\cos(rll))\\
z' &= x{*}(-\sin(yaw){*}\cos(plt))\\
&\quad +y{*}(\sin(plt))\\
&\quad +z{*}(\cos(yaw){*}\cos(plt))
\end{aligned}$$

With the x′, y′, z′ produced from the inverse matrix, azimuth, and elevation may now be calculated for a target on the helmet.

$$\text{Azimuth} = \text{Atan2}\ (x', z')$$

$$\text{Elivation} = \text{Atan2}\ (y', \sqrt{x'^2 + z'^2}$$

THE MODULAR COCKPIT APPROACH TO AIRCREW TRAINING DEVICE DEVELOPMENT

James P. Exter
Manager, Simulation Projects
McDonnell Douglas Helicopter Company, Mesa, Arizona

ABSTRACT

The modular cockpit approach to simulation provides the flexibility to meet varying training device requirements at reduced initial and life-cycle cost across the aircrew training device suite. The full mission, part task, and team training devices all use a common cockpit module that contains all of the hardware and software that is particular to the aircraft (fixed or rotary wing). The current state-of-the-art of microprocessors provides enough computing power to more then meet the computation requirements of today's most sophisticated aircraft systems. Previous size, power, and cooling requirements prohibited one cockpit from meeting both a classroom part-task or familiarization requirement and the full mission training requirement with the same device. McDonnell Douglas Helicopter Company has demonstrated this concept for their Apache Full Mission Simulation system.

Some of the major benefits of the modular cockpit approach include:

1. Common components for various levels of devices therefore reduced logistical support requirements across the training device suite.

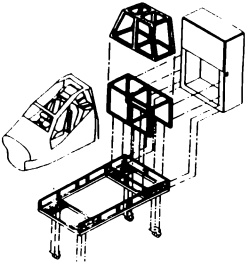
2. Common software across training devices.

3. The ability to use one visual environment to meet various project requirements by connecting different cockpits to the display system.

The modular cockpit approach was developed at McDonnell Douglas Helicopter Company to meet a wide range of engineering, marketing, and training requirements at reduced initial and operating cost. The Modular Cockpit Approach to Aircrew Training Device Development paper discusses the concept, requirements, development, integration, and use of the modular cockpit system.

BACKGROUND

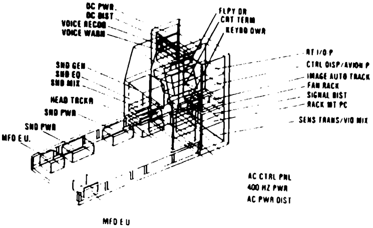
When the Simulation Systems Organization at the then Hughes Helicopter Company was formed in August of 1984 the requirements close at hand seemed overwhelming. The newly formed organization was to assume responsibility for full mission engineering simulation for the Army's next generation helicopter - the LHX. The then Hughes Helicopter Company, now McDonnell Douglas Helicopter Company, simulation organization also had responsibility for the design and proto-typing of the entire LHX aircrew and maintenance training device suite. A handful of senior simulation system engineers studied ways to maximize hardware and software designs across the training device suite to reduce development costs, i.e., time, labor, materials, maintenance, etc. As it turned out, the 1984-85 LHX program schedule requiring the training device suite to be delivered prior to the first flight of the aircraft was just one of a series of false alarms. However, from the initial conceptual discussions emerged the Modular Cockpit approach to simulation. The Modular Cockpit approach has been demonstrated on subsequent



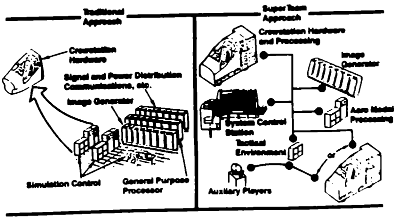
simulation development programs at McDonnell Douglas Helicopter Company and has proved to be an extremely efficient method of developing simulators to support a variety of projects and users. The Modular Cockpit approach for training device support is also applicable where various levels of aircrew training devices i.e. part task, full mission and team training are required. In addition, multiple modular cockpits can be used with one set of supporting simulation resources (visual, IOS, environmental simulation, etc.) in effect giving the user more then one full mission simulation capability. The Modular Cockpit approach is particularly interesting when looked at in a training center context where training for multiple aircraft configurations is required.

THE MODULAR COCKPIT APPROACH

The MDHC Modular Cockpit approach requires all simulated aircraft subsystems peculiar to a specific weapons system be located within the cockpit module structure. The computing resources, real time I/O, graphics engines, power supplies, sound, communications, etc. are all located within the cockpit module. (Fig. 1) This in-cockpit hardware supplies all the computing resources required to execute the software for the cockpit avionics, the flight model, controls and displays, sound, comm, etc. The supporting generic simulation resources that remain outside the cockpit are the out-the-window visual system, instructor/operator station, tactical environment simulation



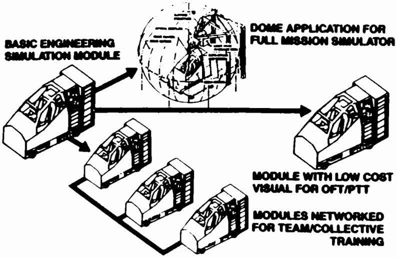
computing systems including intelligent adversary and friendly aircraft control, and of course, the motion system and control loading support equipment if required. The Modular Cockpit approach is a radical departure from the traditional approach to building a full mission simulator in that the supporting facility requirement is much smaller (Figure 2). The cabinets of computers, interfaces, power supplies, power controllers, video switching, etc. no long occupy the raised floor area. Only those simulation systems that are truly generic to all projects are located in the facility. The Modular Cockpit concept works well for a large aerospace company trying to get as large a return on the simulator capital investment as possible. By moving project specific modular cockpits in and out of the costly visual environments, the cost of the visual environment (image generator and display system) can be spread across multiple projects.


Modular Simulation Architecture

The modular concept would also work well at a facility like the U.S. Army aviation training facility at Fort Rucker where multiple aircraft types are simulated for training. Specific visual, motion, IOS, etc. resources could be targeted to the most pressing training need. The modular cockpit approach allows the specialized computing equipment, instructor/operator station, and environmental simulation systems to support multiple projects instead of being fixed for single project simulation support.

As was briefly discussed in the opening background section, one of the principle drivers for the Modular Cockpit approach was the requirement to build multiple aircrew trainers with similar cockpit types. The Modular Cockpit can support all part task, full mission or team training requirements with one hardware and software baseline that can be expanded or contracted to meet the training requirement. In addition, when the full mission cockpit module is extracted from the out-the-window environment it becomes a part task training device capable of training all non-visual tasks.

ADVANCED TRAINING CONCEPTS
EXTENDED APPLICATIONS OF THE MODULAR COCKPIT


BASIC ENGINEERING SIMULATION MODULE
DOME APPLICATION FOR FULL MISSION SIMULATOR
MODULE WITH LOW COST VISUAL FOR OFT/PTT
MODULES NETWORKED FOR TEAM/COLLECTIVE TRAINING

The modular cockpit approach also provides for rapid systems integration. All cockpit interfaces are standardized and plug compatible with the facility visual, IOS and tactical environment simulation resources. Because the modular cockpit is capable of stand alone operation, all on-board systems can be tested prior to integration with the full mission facility resources. Individual subsystem development responsibility can be assigned to individuals or teams and can be worked in parallel with other subsystem developments. This concept of concurrent engineering reduces overall development time and more narrowly focuses subsystem development efforts. The MDHC experience is, modular cockpits are fully integrated and operational in days not months, as is typical for full mission simulation systems.

For the project, the modular cockpit philosophy allows for mission rehearsal in the true sense because multiple modular cockpits can be configured together for team evaluation/training. Helicopters and fighter aircraft typically fly in teams - they should train in teams as well. Obviously this requires additional visual and display resources but many different cockpit combinations can utilize these resources. In addition, air-to-air engagements can also be simulated by using any Modular Cockpit to simulate the enemy ship.

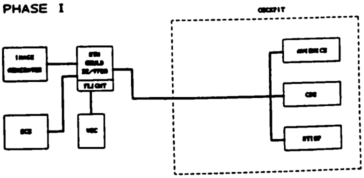
Modular cockpits can be configured quickly with or without visual systems to meet a variety of team training requirements. Without the expensive visual resources, cockpit familiarization, classroom training, part task evaluation, and human factors analysis can all be conducted while other projects utilize the visual environment. Todays simulator also plays a large roll in marketing the company's products. The modular cockpit can easily be moved to trade shows or other facilities to facilitate demonstrations. Modular cockpits are also Simnet/Airnet compatable if larger more robust battlefield scenarios are required.

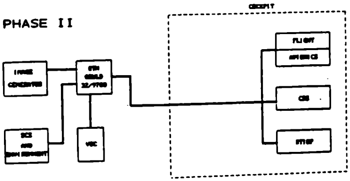
A significant life-cycle cost benefit is realized when one hardware/software approach is implemented across the total family of aircrew training devices. If the avionics software developed to support the Full Mission Simulator also runs on the Operational Flight Trainer, Cockpit Procedures Trainer, Team Trainer and Weapons System Trainer there is a substantial savings in development, integration, documentation, update, and maintenance for the customer. In addition, only one software development facility and Post Development Software Support Facility (PDSSF) is needed. If the same microprocessor based computing resources and graphics engines are used on all the aircrew trainers; again, a substantial savings. If the same cockpit interior panels/instrumentation, power supplies, sound simulation, cockpit shell etc. is used across all the entire aircrew training system the reduction in the logistical support system and therefore life cycle cost is obvious.

MODULAR COCKPIT IMPLEMENTATION

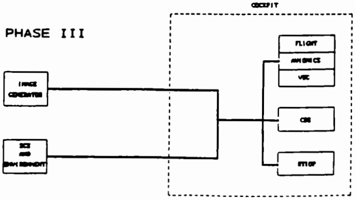
The McDonnell Douglas Helicopter Company's Engineering and Training Simulation Department took a phased implementation approach in developing the modular cockpit. In Phase 1, the real time I/O, avionics systems and controls and displays computing systems were located in the cockpit. The same UNIX-based microprocessor target systems were used for software development. A PSOS real time operating kernel was used to support 15 hz, 30 hz, and 60 hz, operation as required. All the real time code was written in Ada with the exception of some of the interface driver software requiring bit manipulation features not available in the Ada version of software that was first used. (Today, a more developer-friendly software development approach has been implemented. SD and XD Ada software is being developed on a Vax cluster under the VMS operating system and cross compiled to the 68020 single board computer target machines. The download is executed over Ethernet.) In all cases, standard off-the-shelf VME bus circuit boards were used for discrete inputs and output, analog inputs and outputs, the three board graphics sets and the processor and system controller boards. No custom circuit boards were used with the exception of small scaling and lighting control boards which were part of the signal distribution system.

In Phase 2, the FLYRT helicopter flight model was petitioned across three 68020 based single board computers in a separate VME chassis and interfaced with the flight controls and visual system. The migration of the flight model to the microprocessors was a critical event in the modular cockpit evolution because it was always viewed as the single most difficult task for the single-board computer systems.





The Phase 3 tasks involved streamlining the data collection system, re-hosting the visual system, and completing the re-hosting of the tactical environment software. The move to a totally microprocessor-based simulator system is not directly required for cockpit modularity but was precipitated due to the outstanding performance of the cockpit microprocessor systems. The cost savings for simulation applications was determined to be roughly 17 to 1 in favor of the 68020 microprocessor-based single board computing systems over any of the then available super-mini computers.



MODULAR COCKPIT TECHNOLOGY

The heart of the modular cockpit approach is the current generation of microprocessor based single board computers (SBC's) and graphics engines. No longer are super-mini computers required to provide the real-time computing power necessary to support 30-60 HZ simulator operations. MDHC chose the 68020 microprocessor-based single board computer because of the large number of vendors that supported that technology and the availability of Ada compilers. The performance of a distributed 68020 system is adequate for virtually all full mission simulation tasks. A VME bus was selected over other available bus architectures again because of the number of board products available i.e. CPU, I/O, memory, interface, and graphics systems.

Today, multiple racks of I/O interface cabinets are no longer required for full mission simulation systems for two reasons. First, the newer aircraft with mostly glass cockpits no longer have the high I/O channel count once required for older aircraft with analog "steamer gauges". The reduced number of analog guages also decreases the required support electronics i.e., resolver, synchro, analog scaling etc. With efforts to reduce pilot workload have come more efficient ways for crew members to communicate with the aircraft, often reducing the number of discrete cockpit input devices. Second, 64 discrete inputs or outputs or 32 analog inputs or outputs are now located on one circuit card, thus decreasing the "real estate" required to support in-cockpit switchology and instrumentation.

Power supplies as well as power requirements have also been reduced as a result of advances in integrated circuit technology. Switching power supplies with multiple output voltages reduce size and cooling requirements which have already been reduced due to component miniaturization and reduced power requirements. Even linear power sources have benefited from a reduction in component size, weight, and cooling requirements. Sound and communication systems are also being driven by component miniaturization.
The current sound simulation techniques employed are digital sampling and sound synthesis. Both high technology methods require considerably less space and power than previous generation analog sound simulation systems.

The current generation aircraft fly-by-wire aircraft control systems have also played a part in reducing simulator cockpit space and interface requirements. Electronic flight controls have no control surface or other force feedback to the pilots stick and therefore require no electronic/hydraulic/pneumatic loading systems. Most next generation aircraft are proposing fly-by-wire electronic flight control systems because of the benefits associated with computer controlled aircraft control surfaces.

One of the significant advances in technology that directly benefited the modular cockpit approach for simulators is the advancements made in graphics engines. Until recently it has been impossible to find 875 line resolution color graphics engines that could work in real time, except an external sync, and fit in a VME card rack. Today VME form factor graphics engines can support Multi-Function Displays, Helmet Mounted Displays (HMD), Heads-Up Displays (HUD) or any aircraft display system. The VME-based graphic engines work particularly well when used to support the text portion of a text-over-video display system commonly used on todays aircraft.

The use of Ada software is not a requirement for the modular cockpit, however, it is a requirement for the next generation of military aircraft and training devices. In the later phases of the engineering simulation life cycle more and more actual aircraft software is utilized with or without aircraft equipment to validate performance and increase fidelity. Our use of the Ada software language has little effect on the development of the modular cockpit in spite of it's mixed reputation.

CONSTRUCTION OF THE MODULAR COCKPIT

The Modular Cockpit was a top down concept to meet the overwhelming requirement to develop all the training devices for LHX. Similarly, the Modular Cockpit construction requires a top down approach where size, weight, eye point, electronics rack size, cabling, cooling, power requirements and distribution, visual display interface, video, power, computer interface and cockpit fidelity must all be accommodated. Both Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) were used in the Modular Cockpit development process. Although the initial design period took a bit longer before parts were put on order, the reward was no major surprises during subsystems integration. When the first Modular Cockpit was complete and ready for systems integration with visual, tactical environment, and system control station resources, the integration period took less then two weeks.

THE MODULAR COCKPIT FUTURE POTENTIAL

If you refer back to the Phase 3 figure, you will notice the distinct separation between the generic simulation resources and the aircraft weapons systems specific resources that make up the full mission simulator. If the government were to provide for or even set requirements for the generic simulation resources and their interfaces all aircraft weapons system specific simulations could be tested, evaluated or demonstrated in an equivalent environment. Potential ATF, A-12 or LHX contractors could bring their Modular Cockpits to the Government test facility and be evaluated in the same tactical environment using the same performance measurement criteria and the same out-the-window visual and in-cockpit sensor video generation systems. Finally, a one for one comparison capability prior to full scale development and fly-off.

The final phase in modular cockpit development is to make the full mission simulator totally self contained within the cockpit module. Smaller more powerful image generation systems for out-the-window and the sensor video are being introduced every year. Artificial intelligent based instructor/operation capabilities are also being developed. Visual display systems, with the exception of helmet mounted displays, and motion simulation remain the only capabilities that are physically beyond the current modular cockpit technologies.

SUMMARY

The Modular Cockpit approach has significant cost savings potential for government and industry full mission simulation laboratories/training facilities. Expensive visual, IOS and special computing resources can now service more then one project or training requirement. Cockpits can now be easily built to meet part task, full mission, or team training requirements as needed. Buying another full mission simulator capability no longer means buying an image generator, a display system, an Instructor/Operator Station and another tactical environment simulation - just add another modular cockpit.

9-3294-CP

MAN-IN-THE-LOOP SIMULATION AS A DESIGN TOOL

P.H. Cerchie
McDonnell Douglas Helicopter Company
Mesa, Arizona

Abstract

Engineering design begins with the recognition of a need and the conception of ideas to meet this need. It proceeds with the definition of the problem and continues through a program of directed development, which leads to the construction of a prototype which can be evaluated and modified. It concludes when a product is produced that meets the original need.

Engineering design is generally an iterative process, often continuing beyond the point at which manufacturing begins. Correcting problems after the product is fielded means lost revenue and potential damage to a company's reputation.

Simulation has typically entered into the evaluation process late in a program. Consequently, one of engineering's most effective tools is the most under-utilized.

This paper describes a case which utilized simulation during both the design and evaluation phases. Part-task and full man-in-the-loop simulations were run at various points. The objective was to evaluate sensors and to define displays for use in a helicopter obstacle avoidance system. A flight scenario was developed and tested, display concepts developed and evaluated, and sensor models evaluated using simulation as the primary tool. This activity was completed within eight months on a minimal budget.

This paper supports the notion that the iterations leading to an optimal design can be well served by the early use of man-in-the-loop simulation.

Introduction

Tactical helicopter pilots face unique dangers as a result of the special capabilities inherent in the equipment they utilize and the missions they fly. The probability of accidental strikes is high when flying NOE (Nap-of-the-Earth) missions using FLIR-based (Forward Looking Infrared) night vision systems. Experience has shown that wires and supporting towers are not reliably detected by pilots using night vision aids.

The Obstacle Avoidance System (OASYS) Study was designed to evaluate four obstacle detection sensor models and two detection ranges to determine whether any of the sensor/range combinations will help to support safe flight. The goal of the effort is to provide the U.S. Army CECOM Center for Night Vision and Electro-Optics (C2NVEO) with both sensor and pilot performance descriptions from which they can formulate a requirements definition.

This paper describes the use of simulation as a developmental and evaluation tool in support of the OASYS program.

Study Description

The experiment was designed to evaluate the pilot's ability to avoid obstacles, such as wires and towers, not reliably seen on FLIR-based night vision systems. The study objectives made it an ideal application of both partial and full rotorcraft simulation capabilities. Simulation provides an excellent non-life-threatening man-in-the-loop experimental environment. The MDHC AH-64 engineering simulator with both a dome display and IHADSS (Integrated Helmet and Display Sight System) was used as the testbed.

The objective, testing for differences between various combinations of sensor systems and detection ranges, included four sensor models and two detection ranges. These were treated as software systems in the simulator. Actual sensor hardware was not used.

A visual database was designed to support realistic mission conditions, while at the same time affording some control for purposes of accurate data collection. All visual cues were presented on the IHADSS, although a minimal low gray scale out-the-window "dark" representation was displayed on the dome. Experienced pilots felt that a total lack of out-the-window cues is unrealistic, even for night flight.

Sensor Models

The four sensor models are shown in Fig. 1.

	1	2	3	4
Field of View	**6X8**	**20X30**	**30X40**	**30X90**
Frame Time (sec.)	**.2**	**.5**	**.033**	**1.5**
Slewing	**Velocity Tracked**	**Velocity Tracked**	**Head Tracked**	**Fixed Foward**
Stabilization	**Pitch**	**Pitch**	**PNVS**	**Roll/Pitch**

Fig. 1. Sensor Models.

These parameters were provided by the U.S. Army CECOM Center for Night Vision and Electro-Optics. These systems represent characteristics of systems on which data is required to make a determination of the potential to support safe flight. The goal is to gain an understanding of current systems in a realistic, yet safe, test and to begin the process of defining future research directions and formulating a requirements definition.

The velocity tracked sensors point in the velocity path of the aircraft in the azimuth only. While this may not be the ideal for actual operational conditions, the simplification did not influence our ability to test for differences between velocity, head tracked, and fixed forward models. Pitch stabilization, then, means that the sensor is always looking "ahead" regardless of aircraft orientation in pitch. Roll stabilization means that the sensor azimuth is always oriented perpendicular to down.

The models provided perfect information for purposes of the study. If the obstacle was within the detection range, within the sensor field of view, and if the sensor was "looking" at the obstacle in coincidence with the associated frame rate, then a detection occurred. There were zero false alarms.

The process of developing a system as a software model for use in the simulation environment includes several steps. The initial algorithms need to be identified and/or developed and translated into the appropriate code. A method for evaluating and demonstrating proper functioning needs to be determined. Extensive testing throughout the developmental work is conducted to insure that the model is working correctly and is integrated into the total simulation test system. The sensor systems interact with the database, avionics, and data collection systems, and all communications must be verified. Many iterations occur before a system is said to be complete, functioning properly and totally integrated.

Once the code has been developed, early evaluations are conducted by the simulation software engineer using a joystick to 'fly' the systems. Initial checks are made to see how the models are detecting and passing information. As a system progresses, a simulation test pilot is brought into the loop as a subject matter expert. The pilot tests the models in "real" flight, stressing the systems in various flight envelopes. A simulation test pilot supports the software engineer as an evaluator of a system by thoroughly understanding the principles behind the system and evaluating system performance under rigorous flight conditions which are likely to uncover flaws. As problems are discovered, they are corrected and the scheme is repeated until the system is demonstrated to work as intended.

The final configurations are then taken into the full simulation experimental test.

Displays

Initially, a thorough symbology evaluation plan was included in the OASYS study; however, portions of the evaluation were eliminated due to cost/schedule limitations and will be performed at a later date. The completed evaluation will be described, and the follow-up work will be outlined, as it is an important factor for optimizing the operator-system interface.

Once a sensor makes a detection, the information is passed to the avionics system for display. The OASYS symbology consists of two solid cones, shown in Fig. 2. The symbols appear on the IHADSS flight page, as shown, and are easily differentiated from the basic flight symbology. The cones appear as solid white to the pilot.

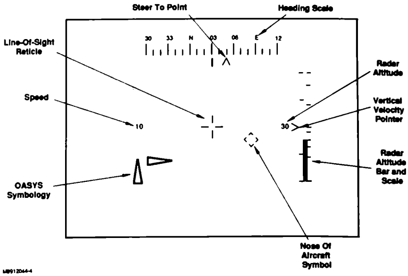


Fig. 2. IHADSS Flight Symbology and OASYS Symbology.

The OASYS symbols are superimposed on the obstacles in the FLIR image, providing a spatial reference for the pilots. The vertical cone indicates a vertical-type obstacle and appears at the the top of the obstacle as a height indicator. The horizontal cone indicates a wire and overlays the wire in the center of the area detected by the sensor. The symbol is maintained on the display until the pilot passes the obstacle in the data base. If the sensor field of view is larger than the PNVS field of view and a detection occurs outside of the PNVS field of view, then the symbology is edge-limited as shown in Fig. 3.

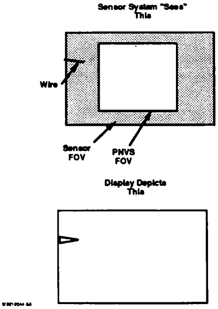


Fig. 3. Edge Limited Symbology.

The abbreviated evaluation of the symbology consisted of generating concepts, choosing the symbol which was closest to the theoretical processing capabilities of the sensor system and would provide a minimum of information to the subjects, and performing some preliminary testing prior to the final experiment.

Symbol concepts were generated by the C2NVEO engineers and by MDHC pilots, avionics and crew systems engineers. The concepts were evaluated on paper in a open forum as to merits for use with an OASYS/IHADSS. The cones were chosen for their simple nature as the best match for the proposed sensor processor. The cones provide location and type (vertical or horizontal) information only, much as the sensor returns do in the unprocessed state. It is not known at this time whether the sensor processors will be able to provide the pilots with additional information.

The cones were integrated into the simulation avionics system for some preliminary testing. Flashing versus steady presentation, size and brightness were investigated prior to finalizing the OASYS symbology.

Initial thoughts were to flash the OASYS symbols in order to provide additional differentiation and attract the pilot's attention. Flashing symbology was flown by four pilots, and all recommended going to steady presentation. The fact that the symbology was spatially referenced and was operating at a 15 HZ update rate caused the symbols to appear excessively jumpy. Steady symbology moved more smoothly and was easier to follow on the display.

The cones were sized twice as large as the existing flight symbology. This differentiated the OASYS cones from the bugs on the basic flight page. Larger cones were deemed unnecessary, and in fact tended to occlude the image of the obstacles.

OASYS symbol brightness was kept consistent with the IHADSS symbology level. Brighter symbols were easy to wash out if the pilot adjusted the display contrast for a very bright image.

While this method of choosing a symbol is fairly common, a more scientific method should be used prior to fielding a system for operational use. The operator-system interface should be thoroughly addressed in the effort to eliminate the possibility of a good design rendered less effective by a sub-optimum display. Consequently, a part-task simulation study is planned as a follow-on to this effort. The original symbol concepts will be evaluated by pilots in the part-task environment in an effort to identify which symbology is preferred/yields the best performance and what the minimum informational requirements are for the performance of a safe avoidance maneuver. The informational requirements of the pilots have not been addressed, yet are the key to providing/designing a usable system, and for determining if available hardware can support an effective display.

Date Base

The data base developed for the study was a canyon with a flat floor about 600 feet across. Walls approximately 200 feet high enclosed the canyon on both sides. The course consisted of 17 segments, each 2 kilometers long, which joined at 30- or 60-degree heading changes. A roadway wound through the canyon for the pilots to follow. The data base was populated with about 975 trees per square kilometer ranging in height from 30 to 70 feet.

Obstacles were placed throughout the course. The basic models are shown in Fig. 4.

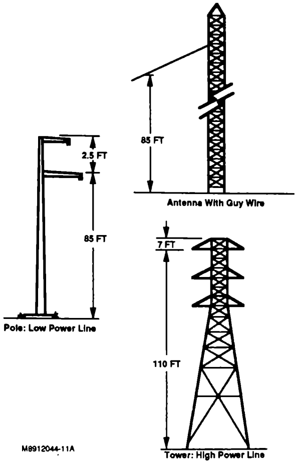


Fig. 4. Basic Obstacle Models.

The pilots were tasked with flying the course at an altitude of 70 feet and an airspeed of 80 kts while following the road. When an OASYS warning was presented, they were to avoid the obstacle using whatever maneuver they preferred.

The pilots flew a total of nine trials: one baseline and eight test.

Complete flight history data and sensor and pilot input data were recorded (including subjective data).

Summary

In the study described above, the early utilization of the simulator for initial evaluations of concepts and procedures eliminated major schedule slips and the need to repeat full simulation sessions once the experiment was started. By exploring ideas in short (1- to 2-hour) sessions and proving that procedures and systems were functional, all of the rework that typically occurs once an experiment begins was eliminated. In addition, the experimental design was refined as evidence arose showing that certain issues were more critical than others. For example, the need to test each sensor model at two detection ranges yielded the requirement for eight test conditions. Initially, the ranges were to be mixed throughout the course. It was determined that this approach did not yield good subjective data as the subjects did not have a good basis for comparison. Using a single range per test session allowed for a clear impression on which to make a judgement.

For the final experiment, data collection was projected for a total of 72 runs, yielding performance data on 1024 obstacles. When the study was completed, only 48 data points had been lost. In addition, the actual simulation time required to perform a full man-in-the-loop simulation experiment was reduced.

Simulation has always been a very powerful evaluation tool once a system or design has been finalized. During the evaluation of a new design, however, flaws or drawbacks are often discovered. This leads to rework and an evaluation of the fix. Using simulation in small doses throughout the development of the pieces of a system or design can help uncover the flaws before beginning the evaluation of the whole system. Entering into the full man-in-the-loop simulation with a pretested design will facilitate a successful and less expensive test.

89-3296-CP

MODULAR AIRCREW SIMULATION SYSTEMS -- 1989 ADVANCES

David E. Powell & Dr. James W. Dille
McDonnell Aircraft Company
A Division of
McDonnell Douglas Corporation

Abstract

Large scale multi-ship manned simulations are increasingly necessary for the development of advanced aircraft systems and for effective aircrew training. This paper discusses the development of technologies to achieve this capability in an affordable manner. These technologies include reconfigurable cockpit systems, the rehosting of full fidelity aircraft simulations onto parallel architecture microcomputer systems, and the development of real time simulation networks.

Introduction

In an age of increasingly sophisticated and expensive aircraft and weapons systems, a requirement exists for large multiple aircrew simulations. To meet this need in a cost effective manner, while at the same time preserving simulation fidelity and performance, new techniques were required.

The Modular Aircrew Simulation System (MASS) was developed to demonstrate the feasibility of using current and emerging technologies such as distributed microcomputers, reconfigurable cockpit displays, and low cost video image generators, projectors and monitors.

Design Approach

To attain the number of participants desired in future multiship simulations, it was clear that a distributed approach was necessary. The elements, or modules, of a large multiship simulation such as fighters and attack aircraft, threat aircraft, ground based threats, and simulation control and monitoring stations, were identified.

Prior to this development, one or two tightly coupled minicomputers had been used to implement all these elements of multiship simulations. In the MASS system, distinct modules would be hosted on separate microcomputer systems, and with the communication requirements between the modules carefully defined, networked together into an overall air battle simulation.

Each of the modules was then further broken down according to its function and computational requirements. For example, the Weapons functions of an aircraft may be separated from other functions such as Flight Dynamics or Physical Cues, forming distinct sub-modules. Then, as performed on the module level, the communications requirements between sub-modules is defined, tying the sub-module functions together as they are executed by the distributed processors of the host microcomputer system.

With the remainder of this paper primarily devoted to the MASS Attacker fighter module, it is important to stress the training goal of such a simulator. The MASS Attacker is designed to provide the high level training of a mission trainer or weapons systems trainer to pilots already familiar with their aircraft. As such, particular care is given to the fidelity of the airframe, radar and avionics models and displays, and the important pilot interfaces of the stick, throttle, and HUD. It was not deemed necessary nor desirable to simulate the exact touch and feel of every switch in the cockpit.

Previous Results and Current Efforts

Previous results had included the construction of a fighter module. This system employed a reconfigurable cockpit using a rear projection system for cockpit display. The simulation was hosted on an Intel Multibus II system using an IBM PC for software development and has at various times employed both Paragon and IRIS image generation systems for cockpit display and out the window scenes. Two such systems had been successfully interconnected and flown against each other using Simulation Network protocols and standard Ethernet hardware.

This paper will describe recent advances in the development of MASS technologies and systems. In total, these advances represent the latest fighter module in the MASS system, an affordable, rapidly reconfigurable, networkable, high fidelity simulator that has been named the MASS Attacker. Specifically, these advances are:

-- The construction of a low cost, mass producible fiberglass shell incorporating direct view high resolution monitors for the main instrument panel display, a generic throttle quadrant, support for side or center stick and removable side panels.

-- The development of a Motorola VME based microcomputer system as a simulation host. This involves a system resident UNIX development environment and tools to load, execute and debug the distributed target code.

-- The development of a sophisticated set of tools and capabilities that comprise the pilot, and instructor/operator interfaces to the simulation.

-- The rehosting of a full fidelity F-15C simulation onto the VME system. While other simulations have been supported on previous microcomputer systems, they had not been simulations with full capability weapons, radar, electronic warfare, etc.

projectors are used to project the images onto three portable screens, arranged to supply a 220 degree field of view to the pilot. (Figure 1)

Head Up Display System

The Head Up Display (HUD) system generates a HUD image which is mixed into the out the window visual scene. The HUD is created by an MVME147 processor driving another IO Inc. IO-GMA25 image generator. The signal is then video mixed with the Paragon center channel output and displayed. Although the Paragon system is capable of displaying a HUD on its own, it was decided to maximize the performance of the Paragon on its primary task of supplying the out the window imagery and to support the HUD on a dedicated processor and image generator.

Simulation Rehosting

Existing software for the high fidelity F15 simulation was written in Fortran and was running on Gould minicomputers. Porting of the software to the UNIX development environment of the MASS Attacker Workstation was accomplished through a standard TCP/IP link. An Absoft Fortran compiler was selected and was able to compile the Fortran routines with a minimum of restructuring made necessary by the use of certain Fortran extensions on the Gould. Software written for the MASS Attacker itself, such as system software and device drivers were written in C.

The rehosting of simulation onto distributed microcomputers was accomplished through a careful functional decomposition of the code, preparing it for implementation on a parallel architecture system. MODSIM protocols were followed as closely as possible for this decomposition and the definition of the variables to be passed between the sub-modules. The MODSIM protocol specifies that the submodule functions are to be located in separate computers and communicate through an FDDI ring. However, for our goal of an affordable and self contained simulator, our submodules are implemented on processor cards within the VME chassis and communicate across the VMEbus.

The VMEbus communication is done in the form of message passing using a highly optimized assembly language program. Following the functional decomposition, messages were defined and set up between the various processor boards in the system. The Instructor/Operator Station processor (IOS) supplies the frame clock, synchronizing the other boards in the system. (Other functions of the IOS processor are discussed below in the section titled Instructor/Operator Interface)

The operation of the system exploits both parallelism and pipelining as information follows a critical path that generally leads from the discrete inputs, through the airframe, weapons and avionics functions, to relative geometry calculations and finally to the cockpit and the out the window displays. The descriptions of the various software submodules of the MASS Attacker shown in Figure 2 are briefly described below.

Flight Station Controls and Display

The MODSIM function of Flight Station is actually accomplished in two parts in the MASS Attacker architecture. An analog input card is used to read the stick and throttle positions, stick and throttle switches and side panel discretes. The display of the flight station information is performed by the Flight Station Display processor driving main instrument panel as discussed above.

Flight Controls, Propulsion, and Flight Dynamics

The three MODSIM functions of Flight Controls, Propulsion, and Flight Dynamics are combined onto two boards in the MASS Attacker, with Propulsion and Flight Controls functions located together on the Flight Controls board. Depending upon the frame rate desired, all three of these sets of functions have occasionally been hosted by a single MVME147. Together, these submodules provide for manual and automatic flight control modes, engine dynamics and thrust, fuel control, weight and balance calculations, atmosphere, aerodynamics, and the equations of motion.

Navigation, Electronic Warfare and Radar

The MODSIM functions of Navigation, Electronic Warfare and Radar are combined onto a single processor board in the MASS Attacker. Radar capabilities include all air to air modes present in the F-15C. The Electronic Warfare capabilities of that aircraft, such as the Radar Warning Receiver (RWR) are also completely implemented.

Weapons

The MODSIM Weapons sub-module is implemented on its own processor board. It provides for the control of guns, missiles and bombs, steering equations, along with armament and stores control.

Physical Cues

The Physical Cues functions of MODSIM is accomplished through the use of a Pentek 4080 sound board, employing mulitiple TI 32020 Digitial Signal Processors. It provides for the creation of the avionics tones and the sounds of engine whine and rumble, aerodynamic hiss, and missile release, among others.

Pilot Interface

A design goal of the MASS Attacker was to allow its use by inexperienced pilots, while at the same time offering performance measurement tools for both pilots and instructors, and debugging and development tools for system operators. The system may be operated in two ways, by pilots entirely through the main instrument panel display and touchscreen interface using a series of simple menus, or through a terminal attached to the Instructor Operator Station board.

The pilot interface is implemented through a series of menus presented directly on the main instrument panel display. Selections are made by touching the appropriate points. At system startup, a username and password are required from the pilot. At that point, he reaches the main pilot menu and is given choices of initial conditions. Initial conditions include the starting position, orientation, and speed of the plane, along with information such as fuel and weapon loads, navigation waypoints, etc.

Upon selecting an initial condition, the instrument panel is drawn and the out the window scene is generated. The pilot now has control of the simulation through several touch points in the upper corners of the main instrument panel display. He may select the modes of reset, freeze, and operate or return to the pilot menu.

Also available to the pilot during flight is a "virtual tape" system. This is the capability to record all or portions of his flight for later playback and analysis by himself and/or an instructor. Tape functions include record, stop, fast forward and rewind, play forward and play reverse. The pause button can be selected at any time, during record to select the interesting portions of a flight, or during playback to more carefully examine the current gauge positions, radar display, etc. Repeated presses of the play forward or reverse buttons puts the playback into a high speed mode, essentially allowing a quicker scan of the recorded flight path. A tape counter displays the current position of the virtual tape.

Upon completing his flight, the flight path information may be saved to the hard disk on the Workstation board. A Tape I/O touchpoint produces a menu which allows the pilot to select a name for his flight and save it, load and review previous flights, or delete saved flights. When the pilot is through with his session, he simply logs off from the main menu.

Certain pilots and all instructor/operators are given "superuser" status and have access to additional functions from the cockpit display. These include calibration routines for the stick, throttle, and touchscreen, routines which allow for the creation of new ICs from the cockpit display, and many others.

Instructor Operator Station Interface

The capabilities of the IOS interface are divided into two aspects, for day to day instructor use, and for system operator and/or programmer use.

Instructor Functions

When an instructor is present during a pilot's flight, several functions are available to him. He can assume control over setting the simulator initial conditions and modes, and can operate the virtual tape system. An instructor may also view screens that clearly display the current status of the simulation, i.e. aircraft and target positions, current missile load, gun rounds expended, radar modes, etc. Other information, such as how many times the pilot took advantage of the tape functions may indicate to the instructor how much effort the pilot is putting into learning from the simulation. These screens are updated constantly and can be redefined and reconfigured easily. He may also change any variable in the simulation if and when desired, allowing him to generate faults and failures.

Operator Functions

Capabilities of the IOS that would be of interest to a system operator or programmer include real time monitoring and setting of both variables and memory and exception handling. The IOS has access to any variable on any board by name. This is a very powerful tool, allowing the value of any simulation variable to be viewed or set, and is an invaluable tool during both debugging and operation.

Symbolic debugging information is obtained from the final target board load files while still within the UNIX Workstation environment. Both the C compiler and the Absoft fortran compiler produce object and executable files in the UNIX Common Object File Format. With the proper options selected, the final executable load files possess both symbol name and address information and source code line number information. A UNIX resident program strips this information from each load file and creates a separate file containing only the symbol and line number information, formatted in a particular manner.

During the load process, as the executable code is loaded to the proper board, this symbol and line number file is loaded to a predetermined location in high memory on the IOS. The IOS is thus able to search through its own memory to find the current location, i.e. global VME address, of any variable on any board. Since the variable type is also known, the value of the variable may be displayed. Having all of this information located in memory instead of on a disk allows for a great deal of speed for these functions.

Variables can be displayed by board, function, or by name. Multi-dimensional arrays are also supported. Combinations of variables on one or more boards may be viewed on a single screen. Sets of variables may be created and saved to and loaded from the UNIX disk, allowing for the customization of variable display screens (This capability also serves as the basis for much of the Instructor performance monitoring functions.)

Line number information is used primarily by the exception handling capabilities of the IOS. When a target board encounters an exception such as a divide by zero, floating point overflow, etc., a message is sent to the IOS containing information as to the cause of the exception and the address where it occurred. The IOS receives this message and notifies the Operator through the IOS interface. When asked to display details of the exception, the IOS is able to display the name of the board on which it occurred, the exact source code line number, the function name and the name of the original source file. Returning to the Workstation environment, an operator can then find that line of code, and return to the IOS to look at the values of pertinent variables in the simulation which had been frozen at the point of the exception. This has proved to be an extremely powerful debugging tool in a distributed processing environment.

Simulation Networking

Reliable high speed communication is needed between the various modules in a distributed air battle simulation. Limitations on the bandwidth of current network hardware and often unavoidable signal propagation delays dictate that careful consideration be given to the hardware designs and software protocols in order to obtain satisfactory performance.

Analyses were performed and through careful definition of the information that must be transmitted between modules, a Simulation Network protocol was developed. Packet definitions were defined in such a way to support the SIMNET type of "dead reckoning" polynomial extrapolation techniques for bandwidth reduction.

The current network media being used is Ethernet and the Simulation Network interface to the MASS Attacker is through a dedicated MVME147 processor board. This board receives state information from the other boards in the system, decides what information is to be broadcast over the Simulation Network, constructs the packets and sends them out. Various packet types include Airframe Packets, Appearance Packets, Kill Request Packets, Radar Packets, and specialized packets for communication with the Test Director module.

At the same time, this board is receiving such packets from other modules, processing them, storing the information and supplying the state of these remote nodes to the simulation when requested. This information is then used by the avionics modules, visuals etc.

In the current stand-alone MASS Attacker configuration, as shown in Figure 3, two fighter modules are linked together, and eight threat aircraft are generated by a single Threat Generator. The Test Director provides for real time command and control of the scenario and a situational awareness, or "God's eye view" of the battle.

The number of participants on the network may be easily increased. Links to existing computer systems such as the Gould mainframes are accomplished through Simulation Network Interfaces. These are essentially small VME systems with a MVME147 running the network code, and a HSD interface to the Gould. It is through interfaces such as these that the MASS Attackers will be integrated into the existing TAC SIM environment at McDonnell Aircraft in St. Louis, currently being used for training by the Air Force.

In order to present a SIMNET compatible interface to the outside world, a gateway node on the network will be utilized. High speed processors at this node will monitor the packet traffic on the local network and format SIMNET compatible packets for broadcast over long haul networks to the rest of a large SIMNET exercise. Similarly, information is received and reformatted into packets compatible with the local network. Ideally, the local network would also be completely SIMNET compatible, but incorporating a gateway node is in any case necessary to perform certain long-haul networking algorithms. A gateway node also allows the local network to be customized to meet various requirements within the local facility while maintaining SIMNET compatibility to the outside.

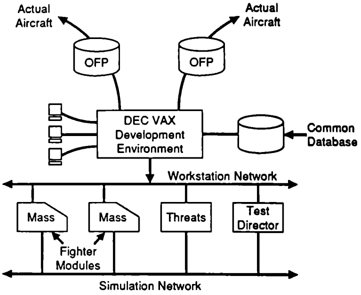


Figure 3: MASS Development Environment

Figure 3 shows several other aspects of MASS system besides the simple standalone configuration and the high speed simulation network. The workstation network, operating under a standard TCP/IP protocol, is the link between the development environments of each module. File transfers and remote file sharing across this network serve to keep MASS system software up to date.

As indicated in the figure, this capability will soon reach well beyond the MASS simulation environment. Large DEC mainframes host the development of the code that will run as the Operational Flight Program on the actual aircraft. This same code, destined for use within a MASS distributed simulation, can then be simply downloaded over the Workstation Network to the run time MASS environment. The excellent development environment of the DEC machines will hold the baseline software configuration. In this way, a modular simulation system can easily be maintained to be as current as the latest aircraft and completely faithful to actual performance.

89-3297-CP

NAL FLIGHT SIMULATOR REAL-TIME COMPUTER SYSTEMS

Akira Watanabe *
Kaoru Wakairo **
Hiroyasu Kawahara ***
National Aerospace Laboratory
7-44-1 Chofu Tokyo, JAPAN

Abstract

NAL (National Aerospace Laboratory in JAPAN) has a flight simulator for research and development of aircraft.

In 1961 the first flight simulator was installed at NAL. An analog computer system was used for real time computation. In 1974 NAL introduced a digital computer system instead of the analog computer. Also in 1980 we installed a new flight simulator to support the development of our STOL research aircraft. Since then we have developed various configurations of the computer system.

First the paper shows the outline of our present flight simulator system which is composed of 4 main pieces of equipment, which are the cockpit system, visual system, motion system, and computer system. The computer system has 6 mini-computers (including two super mini-computers), four of them are used for real-time calculation.

Then, the 6 configurations of the real-time computer system which have been used to the present, are shown. Four of them have not yet existed , the fifth system is being used. The last one is now under consideration. The paper describes each system's features, especially focusing on the synchronization and data transfer among computers.

Last the test items of simulation conducted to the present, are shown.

1. Introduction

The flight simulator is used to simulate the flight dynamics, which includes a human pilot. For this purpose, realtime calculation is indispensable. At present, digital computers are so fast that they are widely used for the main computation part. The flight simulator for research and development requires good operational capabilities.

In 1961 the first flight simulator was installed at NAL (National Aerospace Laboratory in Japan). And an analog computer system was used for real time computation. In 1974 NAL introduced a digital computer system (Ref.1) instead of the analog computer. Also in 1980 we installed a new flight simulator to support the development of our STOL research aircraft, named "ASKA". (The flight tests have just finished this March.) Since then we have developed various configurations of the computer system.

First the paper shows the outline of our present flight simulator system, then describes the characteristics of various real-time computer systems which have been used until the present.

2. NAL Present Flight Simulator System

The NAL flight simulator has been used for research and development (Ref.2), in principle for the following purposes:

① Evaluation of general flight dynamic characteristics
② Evaluation of flight handling quality and stability
③ Evaluation of control system
④ Evaluation of navigation system
⑤ Evaluation of flight instruments and displays
⑥ Evaluation of performance in emergencies and malfunctions
⑦ Evaluation of navigation performance in atmospheric disturbances and gusts
⑧ Evaluation prior to remodeling or repair
⑨ Evaluation of special flight conditions
⑩ Flight training

2.1 Simulator configuration and its characteristics

Figure 1 shows the full configuration of our flight simulator. It consists of the following four main devices:

① cockpit system
② computer generated imagery visual system
③ six degrees-of-freedom motion system
④ computer system for realtime calculation

(1) Cockpit system

The cockpit is identical to the STOL cockpit (Short Take-off and Landing research aircraft which has been developed by NAL), which is the side-by-side type. Its features are outlined in Table 1.

* Senior Researcher
Head of Flight Simulation Lab.
** Researcher
*** Senior Researcher

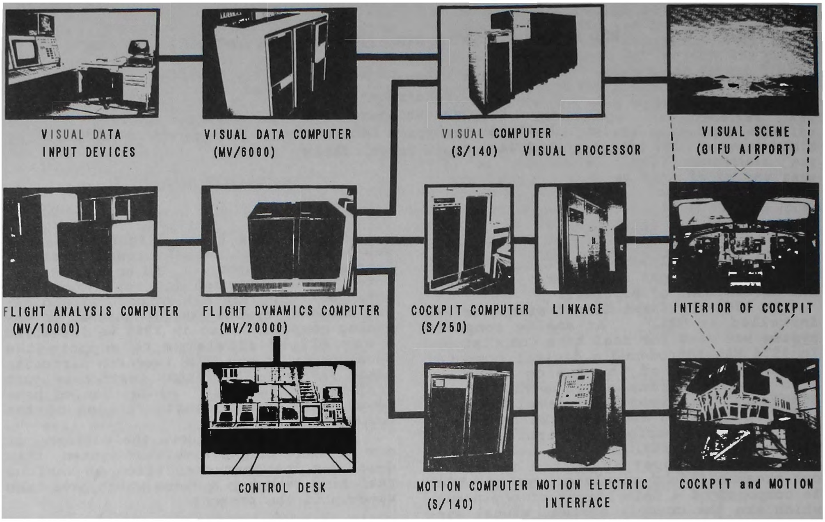

VISUAL DATA INPUT DEVICES
VISUAL DATA COMPUTER (MV/6000)
VISUAL COMPUTER (S/140)
VISUAL PROCESSOR
VISUAL SCENE (GIFU AIRPORT)
FLIGHT ANALYSIS COMPUTER (MV/10000)
FLIGHT DYNAMICS COMPUTER (MV/20000)
COCKPIT COMPUTER (S/250)
LINKAGE
INTERIOR OF COCKPIT
CONTROL DESK
MOTION COMPUTER (S/140)
MOTION ELECTRIC INTERFACE
COCKPIT and MOTION

Fig. 1 Present Flight Simulator Configuration

	ITEMS	CHARACTERISTICS
1	type	CGI
2	number of channels	2
3	number of windows	4
4	CRT	
	ruster	1000 lines
	dot	1300 dots/ruster
	inter race	2:1
	size	26 inches
	color	64 colors
5	number of scene area	100 (max)
6	gaming area	300km × 300km
7	edges	4000 edges
8	light points	2000 points
9	edge intersection	127/ruster (max)
10	occulting levels	128 (max)
11	special effects	smooth shading, smooth fading, lightning
12	moving objects	self + 22 objects
13	frame up date rate	30Hz
14	delay time	50 msec
15	projection way	infinitive display

Table 2 Visual system

	ITEMS	CHARACTERISTICS
1	type	large transport (side by side seat)
2	instrument	conventional, integrated
3	control device	wheel/pedal, side stick control levers
4	feel system	hydraulic digital system
5	visual system	CGI + infinitive display
6	others	sound system on-board control desk

Table 1 Cockpit system

	ITEMS	CHARACTERISTICS
1	type	hydraulic synergetic type
2	degree of freedom	6
3	range of movement	
	translation	± 1m
	rotational angle	± 30°
	acceleration	± 1g
4	position detector	non-touch ultra sonic type
5	payload	~ 8 ton

Table 3 Motion system

(2) Visual system

The visual system is one of CGIs (Computer Generated Imagery). Our system's features are listed in Table 2. For making various three-dimensional visual models, NAL has a program (called IMAP) which allows the generation of visual models through interaction with the digitaizer and the display (Ref. 3).

(3) Motion system

The motion system uses the synergetic 6-degrees-of-freedom type system. The characteristics are listed in Table 3.

2.2. Real Time Computer system (NFCS)

(1) Configuration

Our present flight simulator system has six computers, namely Eclipse MV/20000, MV/10000, MV/6000, S/140, S/250, and S/140. We named the computer system NFCS (Ref. 4, 5).

For realtime calculation , the MV/20000 (execution speed: about 10 MIPS) functions as the center of NFCS. It manages the system for realtime processing and performs the calculations for flight dynamics and flight control. The S/250 controls the cockpit, that is, control devices, flight instruments, and simulated sounds such as those for the engines. One S/140 performs pre-processing for the visual system, the other performs pre-processing for the six degrees-of-freedom motion system.

The MV/10000 makes dynamics and control programs and analyzes results of simulations. The MV/6000 is used to generate image data for the visual system.

(2) Coordination of component computers

The NFCS system is a multi-computer system in which the MV/20000 acts as a central computer, and has distributed memory. Users can select any computer including the MV/20000, according to their simulation job. Unnecessary computers can be disconnected by software.

(3) Realtime management

Overall realtime management is performed by the MSCP (Simulation Control Program) under the control of the MV/20000's AOS/VS (operating system). The MV/20000 serves as a master, and the other computers serve as slaves. These computers are loosely coupled. The basic cycle time is 10 ms. A mode synchronizing signal is sent through the coupling bus (BMC). Time synchronization is not conducted.

(4) Realtime data transfer

The system can transfer data at 10 MByte/sec every 10 msec. Data is transferred from/to the MV/20000 to/from other computers.

(5) Procedure for realtime simulation

A simulation is performed in three stages: ① preparation, ② execution, and ③ post-processing. Figure 2 shows the tasks for each step and the programs in use.

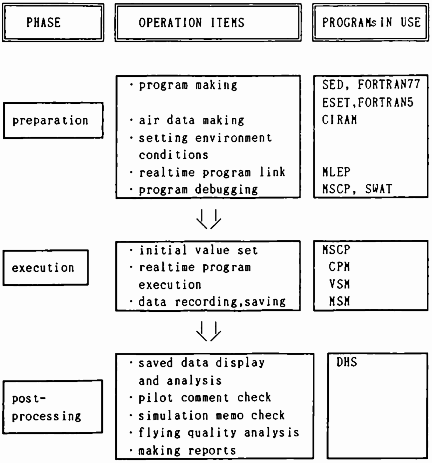
- Preparation -

As for this step, the NFCS system uses utilities software (SED, FORTRAN77, ESET, FORTRAN5, etc.) offered by computer manufacturers and a multi-dimensional (up to four dimensional) function data generation program (CIRAM) developed by NAL. CIRAM employs a table lookup method and a high speed calculations technique (Ref. 6).

- Execution -

Operation in this stage is quite simple. Once the MSCP is started from a terminal, it is only necessary to press function switches on the console of the control desk. These switches are assigned the MSCP functions.

Data obtained during tests can be output to a pen recorder and be stored as digital data on disks.



PHASE	OPERATION ITEMS	PROGRAMs IN USE
preparation	· program making · air data making · setting environment conditions · realtime program link · program debugging	SED, FORTRAN77 ESET,FORTRAN5 CIRAM MLEP MSCP, SWAT
execution	· initial value set · realtime program execution · data recording,saving	MSCP CPM VSM MSM
post-processing	· saved data display and analysis · pilot comment check · simulation memo check · flying quality analysis · making reports	DHS

Fig. 2 Operating procedure and software

- Post-processing -

Digital data is plotted on the graphic display for analysis. The overall characteristics of the simulated aircraft are reported.

(6) Program scheme for flight simulation

In order to extend flight simulation applications for general purposes, NAL is attempting to develop programs that can be used for a wide range of simulations. At present, the programs listed in Figure 3 are available. FSPK-II is the program for the flight dynamics calculation of the aircraft (Ref. 7), CIRAM is used to generate the calculation program of aerodynamic data, and the BAC program is used for non realtime analysis. The results obtained by BAC are compared with the results of the realtime simulations.

In the STOL simulation, about 300 Kbytes of memory are used for dynamics and other control programs, and 1 Mbytes of memory are used for aerodynamic data and engine data. Realtime calculations are performed at 40 msec intervals.

3. Real time digital computer systems

In the previous section we showed our present flight simulator system. Until this configuration we have experimented with various systems. The main reasons for this are as follows :

(1) to cope with the increase of the simulation scale
(2) to make the operation of the flight simulator easier
(3) to correspond with new simulation technologies

Furthermore, the rapid progress of computer technologies is very helpful in supporting the modification of our computer system.

Figure 4 shows the configuration of each computer system which has been used to the present (from FSK-II to NFCS modified). The upper 2 systems (FSK-II family) use 5 M70 minicomputers (made by Mitsubishi Electric Corporation) and the lower 4 systems (NFCS family) utilize an NDG (Nippon Data General) super minicomputer as the master computer. System (f) is under modification.

When the FSK-II was introduced, the calculation techniques improved drastically, because digital techniques were adopted. But as its 'computer was a 16-bit fixed-point calculation type, it was not easy to make simulation programs. In 1980 we renewed the flight simulator computer system (we call it NFCS) and began to use NDG computers. At this time, the users began to use the Fortran language and floating-point data, making it very easy for them to make programs. Thus the simulation scale has been rapidly enlarged. And because the users have repeatedly requested more high processing ability, we have renewed the real time computer system several times.

The characteristics of each system are presented in Table 4. The remarkable points will be described in the following section.

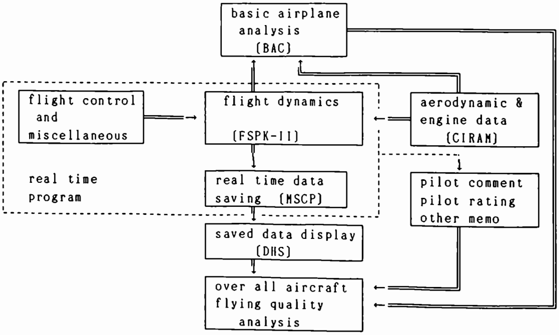

basic airplane analysis (BAC)
flight control and miscellaneous
flight dynamics (FSPK-II)
aerodynamic & engine data (CIRAM)
real time program
real time data saving (MSCP)
pilot comment
pilot rating
other memo
saved data display (DHS)
over all aircraft flying quality analysis

Fig. 3 Software schematics for flight simulation

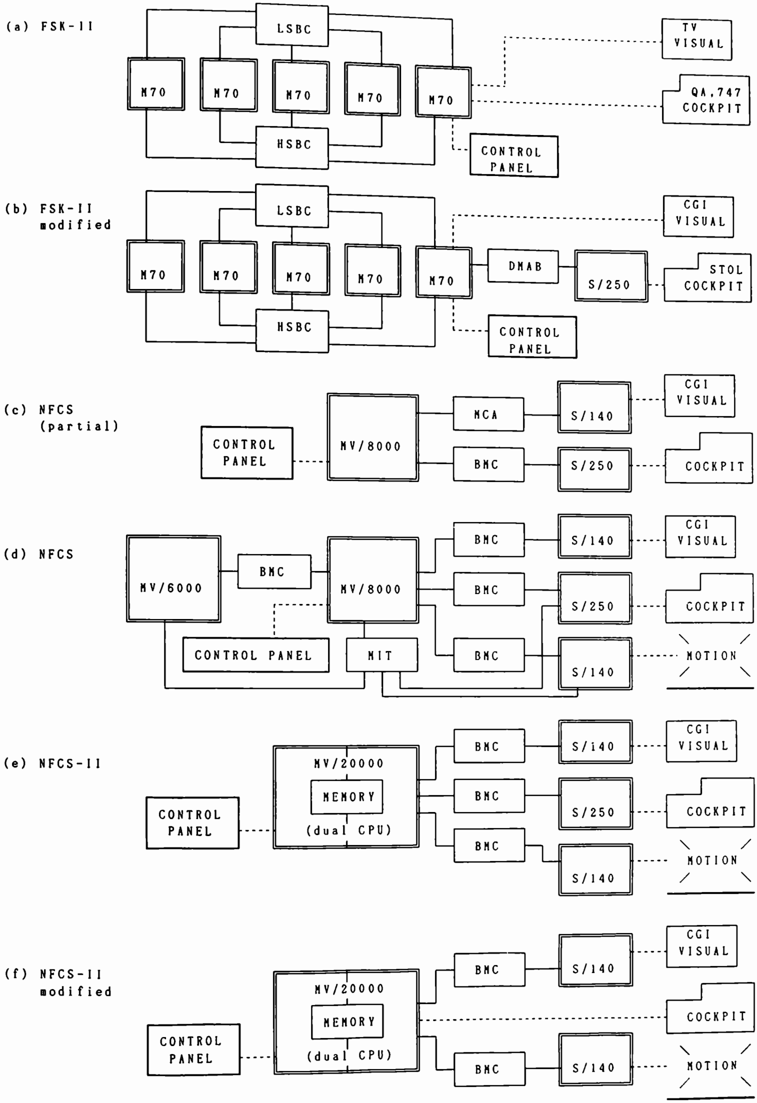


Fig. 4 Block diagrams of the real-time computer systems

3.1 FSK-II

This system is shown in Figure 4(a), it was the first system settled at NAL in 1974 This system has 5 M70 mini-computers which are coupled by two data buses, the LSBC and the HSBC (Ref. 8). The LSBC (Low Speed Bus Coupler) is used for real time synchronization and computer status information transfer. The HSBC (High Speed Bus Coupler) is used for data transfer. Its transfer rate is 1.21μ sec/w. The transfer is made using a broad-cast method.

In order to detect the end of transfer, the interrupt signal was utilized, it took much time (30% of basic period 10msec) and became a great problem.

As for language, there was nothing but the assembler, and the calculation was done by the fix-point method. Then the special higher-order language RTSL (Real Time System Language) was developed. It resembles the PL/1. Nevertheless it was very difficult to make a long program.

3.2. FSK-II modified

In 1978, at the time when the STOL research aircraft project began, the simulator's renewed plan had also started. At the first stage, the cockpit which is nearly identical to the STOL cockpit (side-by-side type), was made. As the function in the cockpit was increased, one computer was added. Among various computers, we chose the NDG computer (Eclipse S/250) for the following reasons.

① We had been familiarized with NDG computers.

② It has a realtime processing ability.

③ As NDG has a factory in Japan, they could correspond to make a special hardware, and to repair in a short time when problems occurred.

④ The computers are general purpose, and they are very useful for scientific calculation.

The system configuration is shown in Figure 4(b), having the following features.

	systems		synchro devices		data transfer devices	coupling methods	OS, real-time monitors	languages
	name	computer	time	mode				
a	FSK-II	M70×5	LSBC	LSBC	HSBC	tight	SCP	RTSL
b	FSK-II modified	M70×5 S250	LSBC (RTC)	LSBC DMAB	HSBC DMAB	tight loose	SCP RDOS + CPM	RTSL FORTRAN5
c	NFCS (partial)	MV/8000 S/140 S/250	(RTC)	BMC	BMC MCA	independent, mode couple	AOS/VS + MSCP RDOS + VSM RDOS + CPM	FORTRAN77 FORTRAN5
d	NFCS	MV/8000 MV/6000 S/140 S/250 S/140	MIT	BMC	BMC	master clock independent, mode couple	AOS/VS + MSCP AOS/VS + M2M RDOS + VSM + CPM + MSM	FORTRAN77 " FORTRAN5 "
e	NFCS-II	MV/20000 S/140 S/250 S/140	(RTC)	BMC	BMC	independent, mode couple	AOS/VS + MSCP2 RDOS + VSM2 RDOS + CPM RDOS + MSM	FORTRAN77 FORTRAN5 "
f	NFCS-II modified	MV/20000 S/140 S/140	(RTC)	BMC	BMC	independent, mode couple	AOS/VS + MSCP2 RDOS + VSM2 RDOS + MSM	FORTRAN77 FORTRAN5

LSBC	: Low Speed Bus Coupler
HSBC	: High Speed Bus Coupler
RTC	: Real Time Clock
DMAB	: DMA (Direct Memory Access) Buffer Coupler
MIT	: Multi Interrupt Timer
BMC	: Burst Multiplexer Communication Coupler
MCA	: Multi Communication Adaptor

SCP	: System Control Program
RTSL	: Real Time System Language
RDOS	: Real Time Disk Operating System
CPM	: Cockpit System Monitor
VSM	: Visual System Monitor
MSM	: Motion System Monitor

Table 4 Characteristics of the real-time computer systems

(1) Coupling

Among M70s the coupling was the same as previously. The S/250 and the M70 are a different type computer, but they have a DMA channel, so we utilized this function. The data goes from one to the other through this line. Its transfer rate is 3.7 μ sec/w. We named it DMAB.

(2) Data transfer

The data transfer between the S/250 and the M70 is controlled by the SCP(Simulation Control Program) in one of the M70s. First the address and the number of transfer data are set in the interface of the DMAB , then the SCP triggers the transfer every 10msec. The end of transfer is detected by the interrupt signal, but the interrupt process of the M70 needed much time , so we tested a non-interrupt technique and it succeeded to reduce the system overhead. The data from the S/250 is transferred by the broadcast transfer method through the HSBC to the other M70 computers.

(3) Synchronization

The real time synchronization among M70s is done through LSBC , but between the S/250 and the M70, we did not adopt a real time synchronization. The real time clock of the S/250 operates independently from the M70's and controls only that computer.

The mode of computer execution (reset, operate and hold) needs to coincide. It is possible to wait a few moments (2-5 periods : 20-50msec) to check the mode. We adopted the following method. The SCP in the M70 checks every 10 msec the function panel which is connected to the M70. The other M70 computers are notified through the LSBC. To the S/250 this mode signal is sent as a part of the transfer data through the DMAB. In the S/250, every 10msec the real time monitor checks the mode signal and matches the self mode to it. It sends its result to the SCP as a part of the transfer data via the DMAB.

The SCP checks the mode sent every 10msec, if the result is good in 5 periods(50msec), the SCP knows that the S/250 has changed to the specified mode. This is a poling method which does not need much time, making it good for a real time system.

3.3 NFCS (partial)

In 1982 we introduced the CGI type visual system. At that time we renewed the computer system of our flight simulator. This configuration is shown in Figure 4(c). We chose the NDG MV/8000 as a main computer. This computer's ability is about 1.3 MIPS. All calculation is done by floating point processing. The previous system (5 M70s and an S/250) was about 1.4 (M70 is a fixed-point calculation type). This showed an insufficient capability, therefore we planned to add another computer with our next modification.

(1) Real time management

The MV/8000's OS (Operating System, called AOS/VS), is the system available for general purpose use. It is not necessarily effective for real time operation. It takes about 3.4 msec until a user's program starts after the real time clock interrupt was input, so we were afraid of injuring the real time operation. As a counterplan , it was proposed to use AOS/RT32 which is made as a real time OS. But this OS is weak in handling files, thus we did not utilize it.

We made a special real time management program called MSCP. This program works under AOS/VS. The S/250 and the S/140 utilize an RDOS (Real Time Disk Operating System) and a special real-time-monitor. These two real-time-monitors were developed by NAL.

(2) Coupling

As it is important to shorten the transfer time , we developed a special coupling device (We named it BMC). Its transfer speed is 8 MByte/sec. It is used between the MV/8000 and the S/250. Between the S/140 and the MV/8000 the standard transfer line was used. Its transfer rate was 300 KByte/sec.

(3) Data transfer

In order to minimize the starting time to BMC, once triggered, BMC repeats the transfer every 10 msec until an order of stop is issued or an error happens. This is very effective in order to minimize a system overhead.

(4) Synchronization

Real time operation is done by a self contained real time clock. Then a real-time synchronization job among computers is not necessary, but it needs a mode synchronization. We adopted a way that a mode flag is sent as a part of the data through the BMC, and the response is received as a part of the data sent from the other computers.

3.4 NFCS

This system is shown in Figure 4(d). Among the NFCS family this system is the most complicated. For the real time calculation two super-minicomputers (MV/8000 and MV/6000) are used. For the visual system the S/140 is used, and for the cockpit system the S/250 is used. Furthermore, for the motion system we use another S/140. Totally 5 minicomputers are connected. This system was established in 1984. For the real time synchronization we tried to use MIT (Multi Interrupt Timer).

(1) Coupling

For this system we have utilized the BMC for all of the transfer. Its transfer rate has increased to 10 MByte/sec. The MV/8000 is a main computer and the others are connected to it by using BMC and MIT.

(2) Data transfer

The data transfer is conducted each 10 msec by using BMC. It takes about 300μ sec, but compared to the task execution cycle of 40 msec, this is negligible. It does not influence the real time processing.

(3) Real time control

Here we tested a master timer method. In Figure 4(d), MIT is the master timer which controls all computers. This timer is controlled by the program (MSCP) in the MV/8000. It has delay circuits in each output to adjust the timing in order to minimize the transfer delay. We tested the adjustment but it was not useful because the user's program was always different, so each time an execution timing was different, it was very difficult to adjust the timing. Therefore this idea was not successful.

(4) Synchronization

The mode synchronization is necessary, thus the mode flag is sent from the MV/20000 to the other computers as a part of the data. And the response sent via the BMC from the other computers, is checked every 10msec. If a response of the same mode is gotten in 50 msec, it is good.

3.5 NFCS-II

At this time, as indicated in Figure 4(e), in order to increase the ability of the computer system, we introduced an MV/20000 (dual CPU type) and we released the MV/8000 and the MV/6000. The others are the same, and the coupling devices are also the same. Furthermore, we connected the MV/10000 and the MV/6000 to the MV/20000 by MCA. The MV/10000 is used for making and debugging simulation programs and for analyzing the simulation results. The MV/6000 is used for making the visual data. The real hardware has already been shown in Figure 2.

It was expected to increase the calculation ability by about 5 times. But every 10msec the real time interrupted signal is checked and its processing takes much time (1.3msec). Thus the ability is not increased so much.

(1) Coupling

The coupling is the same as the previous system. We ceased using MIT. Each self contained RTC is used in stead.

(2) Real time execution

The MV/20000 has two CPUs. The AOS/VS as well as the MSCP works in the CPU0. The special small real time monitor works in the CPU1. User's program are properly separated and they execute in each CPU. The S/140 utilizes an RDOS and the special real time monitor operates under RDOS.

(3) Synchronization

Among the computers, synchronization is the same as previously described. The MV/20000 has two CPUs. Between the CPU0 and the CPU1 the information is exchanged via the common memory. The common memory is also used for the synchronization of these CPUs.

3.6 NFCS-II modified

This system is indicated in Figure 4(f) and is now under modification. The MV/20000 will undertake all of the S/250 tasks. This means that the cockpit signal enters directly in the MV/20000 and the instrument signals are output to the cockpit system. We expect to reduce the delay time. We will report this system's performance at a later date.

4. Evaluation of computer systems

To the present we have conducted various simulation tests using the flight simulator. The results are as follows :

(1) As it is possible to do data transfer by a non-order and independent repeat method, time to start and to end the transfer hardware, is unnecessary. This decreases the time of the system overhead.
(2) As the computers are functionally separated, it is necessary to send only the data needed to a certain part.
(3) It is possible not to do time synchronization among computers. Therefore it does not need the hardware for the synchronization and the job of time synchronization.
(4) The real time accuracy depends on each real time clock. As one simulation time is about one hour at most, this method is not an influence on a simulation.
(5) In order to decrease the time of system delay, the data transfer cycle is several times larger than the task execution cycle.
(6) To establish the mode synchronization, we can wait some time (around 100 msec), compared to the time of human operation (300-500 msec). Thus we indicated that it is possible to use the data transfer line for it.

5. Conclusion

NAL has a flight simulator for research and development of aircraft and recently it is also used for a simulation of spaceplane. Here we discussed our flight simulator especially concerning the real time computer systems.

The FSK-II system was first introduced at NAL, its coupling system has had many experimental faces. Some improvements had been requested. NFCS is comparatively a simple configuration, but the simulation capability is very high and it operates very good.

To the present we have conducted various simulation tests listed in Table 5. Each simulation was finished with expected results.

Acknowledgments

We would like to thank Mr. M. Okabe and Mr. Y. Horikawa of NAL for their help and guidance, and Mr. T. Bando who made the FSPK-II flight dynamics program, for his great effort. Also we make a grateful acknowledgment for NDG (Nippon Data General), JRC (Japan Radio Company Ltd.) and MPC (Mitsubishi Precision Company Ltd.) for their contribution in the developing hardware and software of our flight simulator.

	simulation subjects	test items
1	STOL research aircraft development project	① Evaluation of flying quality ② Evaluation of SCAS control law ③ Evaluation of engine failure and control devices runaway ④ Coupling tests of on-board devices ⑤ Crew trainning
2	YXX (next generation aircaft) design	① Evaluation of FBW control law ② Evaluation of side stick
3	Advanced display	① Evaluation of multi-function CRT displays ② Evaluation of liquid crystal flat panel displays ③ Evaluation of a digital map display ④ Research of simple integrated displays
4	Space plane research	① Evaluation of flying quality in the phase of approach and landing ② Evaluation of landing aids ③ rendez-vous docking
5	Space station	① Design check of space stations ② Evaluation of antenna plorongation
6	miscellaneous	① Evaluation of data buses ② Evaluation of a terminal area guidance expert system

Table 5 Simulation tests

References

1) Y. Horikawa et al. : NAL Multi-computer system (FSK-II) for real time simulation, Information Processing Society of Japan, Paper of Computer Architecture Meeting, 1975. 3.

2) A. Watanabe et al. : NAL flight simulator for research and development, Proc. of the USA-JAPAN Symposium on Flexible Automa tion, ASME & ISCIC, 1988. 7.

3) K. Wakairo et al. : The development of the Interactive Model Assembly Program (IMAP) for a flight simulator, NAL TR (soon published).

4) A. Watanabe:Flight simulator facilities, NDG User's Group Conference, 1984. 6.

5) A. Watanabe : NAL Real-time multi computer system for STOL research airplane flight simulator, Information Processing Society of Japan Annual Meeting, 1984. 3.

6) A. Watanabe : Fast Calculation Method for Table-Lookup Data, Short Paper of Society of Instrument and Control Engineering of Japan, Vol. 21, No. 1, 1985. 1.

7) T. Bando and A. Watanabe : Flight Simulation Program part I (Contents of Program), NAL TR-702, 1982. 2.

8) K. Harada : Data transfer and its con trol of multi-computer system FSK-II for simulation, NAL TR-511, 1977. 8.

89-3298-CP

AIAA Flight Simulation Technology Conference

15 August 1989

"SHARED-MEMORY NETWORKING ARCHITECTURES--SIMPLICITY AND ELEGANCE"

by

Mr. W. Lynn Trainor
Mr. Gary G. Warden
SYSTRAN Corp.

ABSTRACT

Connecting together multiple computers to solve a single real-time problem, such as aircraft simulation, is certainly a very challenging engineering task. This paper addresses some of the unique requirements of distributed computing architectures for simulators, and it summarizes the two major approaches which have been used to date. The concept of a new approach--shared-memory networking--is introduced and compared to the traditional approaches. Next, the design and performance parameters of the first comprehensive shared-memory network implementation are presented. Details of this new design are presented along with implementation considerations.

INTRODUCTION

Background

Computational systems for the real-time aircraft simulation domain have always presented some of the most demanding requirements of all real-time applications. Algorithm computational loads are extremely high with the complexity of today's aircraft subsystems and with the high speed avionics which must be simulated. Added to this is the ever increasing performance envelops of modern aircraft platforms. This increasing performance compounds the computational tasks by increasing the iteration rates on many models. Furthermore, many of these high-rate models are the more complex ones and most difficult to compute.

To organize the problem space for discussion purposes, refer to Figure 1. Aircraft simulators can be simplistically classified in two categories--"simple" simulators and "complex" simulators. Simple simulators are those where only one computer is needed for the computational load. Here the "system architect's" job is rather straightforward. All the system inputs (Figure 2), outputs and all inter-task data reside in one computer, and there are no inter-task communication problems.

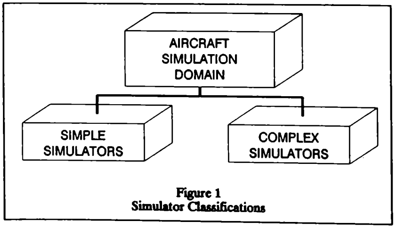


Figure 1
Simulator Classifications

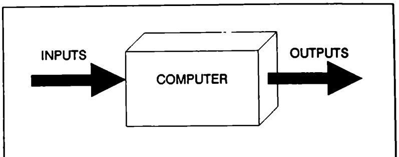


Figure 2
Simple Simulator Diagram

Unfortunately, most aircraft simulators do not "fit" within this "simple" simulator category. Instead, most require multiple Central Processing Units (CPUs) to handle the massive algorithm loading and to take advantage of functionally specialized computers. To obtain the computational power needed, simulator system architects have turned to using distributed computing systems composed of numerous CPUs--all connected with various real-time communication devices (Figure 3). This approach has permitted the system architect to add computer power in "chunks" as needed, and to mix-and-match specialized computers that best match the particular computational problem. For example, use high speed CPUs for algorithmic modules, high speed graphics engines for visuals, etc.

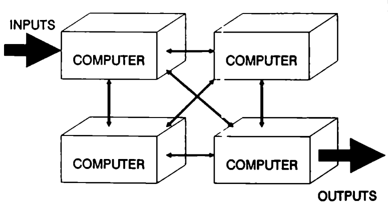


Figure 3
Complex Simulator Diagram

However, these distributed computing systems have presented some very challenging problems in themselves. Probably the most demanding of these has been how to interconnect these distributed computers in a fashion such that inter-processor communications do not adversely affect the computers themselves. Various schemes have been used, and each of these has its requisite strengths and weaknesses.

Published by the American Institute of Aeronautics and Astronautics, Inc. with permission.

Traditional Connection Approaches

Basically two approaches have been used--common-memory architectures (Figure 4) and message-passing networks (Figure 5). The common-memory architecture utilizes a high speed parallel memory bus through which all computers can access a common-memory module (a variant of this structure utilizes a multiported memory in lieu of the common bus.) This architecture has offered the requisite data communications speed along with other benefits:

- virtually no software overhead is needed for data communication,
- fast system reconfiguration,
- fast error recovery, and
- "tight" control over system resources.

However, the common-memory architecture has its limitations. Specifically:

- common-memory bus contention,
- limited physical separation distance between computers (e.g., 50 feet),
- typically must use a single vendor's hardware, and
- can only connect a few computers (e.g., 5 to 10).

The second traditional approach used has been the message-passing local area network (LAN). This architecture uses a serial communication link through which all computers pass message packages. This LAN approach has helped system architects overcome several of the limitations of the common-memory approach. Specifically, LANs:

- allow for the integration of different vendors' CPU hardware,
- allow for a much larger number of CPUs (e.g., tens or hundreds), and
- permit the physical separation of CPUs to distances of hundreds or thousands of meters.

These LAN advantages are of great importance in some simulation applications.

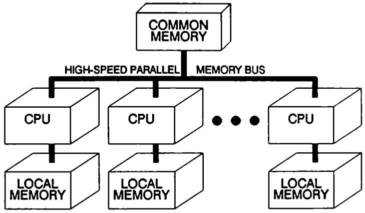


Figure 4
Common-Memory Architecture

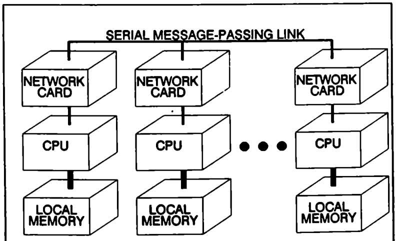


Figure 5
Message-Passing Network Architecture

However, the use of LANs has given the system architect some severe system design constraints, and these are intolerable in many simulator applications. Some of the major LAN problems are:

- data communications are several orders-of-magnitude slower than for common-memory approaches,
- large software overheads are encountered, thus "eating up" valuable CPU time,
- interrupting other computers and maintaining "tight" control of the distributed computer system is extremely difficult over a message-passing LAN, and
- error recovery and system reconfiguration are difficult and time-consuming.

To summarize, the common-memory and message-passing LAN approaches each has its strengths and weaknesses for aircraft simulators. In an area where one approach is good, the other tends to be poor, and vice versa. Table 1 shows this summary comparison.

Table 1
Traditional Approach Comparison

Feature Required	Best Architecture	
	Common-Memory	Message-Passing LAN
High Speed Inter-Processor Communications	X	
Tight System Control Possible	X	
Use of Multiple Vendors' CPUs		X
Allows Large Number of CPUs		X
Allows Physical Separation		X
Low Software Overhead	X	
Fast Error Recovery	X	

Shared-Memory Network Alternative

Again referring to Table 1, you can quickly see that the two traditional communication approaches are complementary. What is really needed is a third alternative for the real-time system architect to use--one which possesses the combined strengths of the two traditional approaches. This alternative would possess the favorable attributes of common-memory--data communication speed within microseconds, tight system control by passing interrupts in microseconds, virtually zero software overhead to pass data or control, recovery from communication errors within microseconds--and also the favorable attributes of the message-passing network--ability to integrate different computer vendors' CPUs, connection of tens (or even hundreds) of computers into one coordinated simulation system, and the ability to physically separate the computers into separate areas or buildings which may be hundreds of feet (or even miles) apart.

A new communication technology has been recently developed which does precisely this. It offers the strengths of both the common-memory architecture and the message-passing network, yet it retains none of the inherent disadvantages of either. This approach--termed shared-memory networking--appears to the software designer as if he is using a common-memory system, yet it is actually a serial-ring network using replicated memory modules at each node.

Recent advances in several technologies have made this new approach feasible. Primarily these are the use of serial fiber optic links operating at very high data rates, and the use of extremely fast and dense electronic parts which enable the compact designs needed.

A shared-memory network is a rather simple concept, and much of its elegance and power derive from this simplicity. Also known as "reflective memories" and "memory networks," these designs operate by placing a network card in each computer to be connected--just like a message-passing network. The cards then communicate over a serial ring architecture. However, this is where the similarity to message-passing networks stops.

Shared-memory networks do not pack multiple data words into messages, attach lengthy protocol information, or pass these complex messages with cumbersome hardware and software protocols. Shared-memory networks do not use the seven layer ISO implementation. Shared-memory networks obviate the need for these cumbersome, time-consuming, and non-deterministic protocols found in message-passing networks.

In a shared-memory network, a datum is communicated very quickly and directly--the transmitting CPU simply makes a high-level language statement: A = B, where the variable A is located in the shared-memory window of the network card. Within microseconds all other CPUs on the network have the same value of the variable A in their shared-memory areas. The network card does this by passing both the datum and the datum address over the network, and the variable is located at the same relative shared-memory address in each computer's shared-memory area.

Control--another critical performance requirement of simulators--can be passed over a shared-memory network by the same mechanisms as data are. Interrupts can be transmitted to all (or selected CPUs) within microseconds. For example, on a ten node network all ten CPUs can be interrupted in less than seven microseconds. Likewise, a new datum value can be transmitted to all ten CPUs in the same maximum of seven microseconds.

This shared-memory network approach also provides a very deterministic performance envelop for the system designer. Control and data communications take place in a few microseconds for an upperbounds, and there are no non-deterministic software routines which must be invoked to complicate this process. Also, using the ring network with short, fixed-length transmissions there are no "unknowns" in the transmission link transport times.

In short, the shared-memory network offers a new "tool" to the simulator architect. This tool is one where all the advantages of a common-memory approach (see Table 2) can be combined with the best features of traditional LANs. The result is a tool specifically optimized for the aircraft simulator domain.

THE SCRAMNET^{TM} NETWORK IMPLEMENTATION

Overview

The Shared Common Random Access Memory Network (SCRAMNet Network) represents the first commercial implementation of a truly general-purpose shared-memory LAN. To date, there have been several special-purpose designs implemented by various companies, each targeted to very narrow sets of requirements. Typically these have been "point designs" for specialized projects or for a single customer's needs. The results of these designs have been very good, and the overall system performance of these specialized systems has been excellent.

However, these point designs--because they have been so "customer specific"--have not been made broadly available to the engineering industry. Thus, what has been lacking to date is a truly general-purpose shared-memory LAN; one designed to satisfy a large segment of the real-time industry.

The SCRAMNet Network design has evolved from this historical perspective. SYSTRAN has been active in developing shared-memory networking technology to meet the unique needs of customers, and has previously produced two shared-memory networks. Table 3 is a summary comparison of these two prior networking systems, along with the current SCRAMNet Network design.

As a result of this prior design experience and the real-time engineering community's response to these shared-memory networking results, SYSTRAN undertook an effort in January, 1988, to design and develop a shared-memory networking system that would satisfy the needs of a broad segment of the real-time community.

The result, the SCRAMNet Network design baseline, was formulated from this broad segment of the real-time community, with aircraft simulation being the most heavily considered requirements. Thus, the SCRAMNet Network design is a flexible design, one containing a full range of functionality and features needed for real-time networking, and one having a large enough customer base to make this new design commercially viable. Although it is not a general-purpose LAN in the sense that it is intended to replace message-passing LANs in the non-real-time market, it is indeed a general-purpose LAN in the real-time domain. It is specifically optimized for the real-time domain, and this domain only. Thus, its real-time performance is clearly unmatched by other approaches.

Table 2
All Three Alternatives Compared

Feature Required	Best Architecture		
	Common-Memory	Message-Passing LAN	Shared-Memory LAN
High Speed Inter-Processor Communications	X		X
Tight System Control Possible	X		X
Use of Multiple Vendors' CPUs		X	X
Allows Large Number of CPUs		X	X
Allows Physical Separation		X	X
Low Software Overhead	X		X
Fast Error Recovery	X		X

Table 3
SYSTRAN Shared-Memory Network Designs

Feature/ Parameter	Design		
	Design A	Design B	SCRAMNet
Physical Size	5 Separate Cards	2 Separate Cards	1 Card Assembly
Link Medium	Wire	Wire	Fiber Optic
Link Bandwidth	20 MBit	50 MBit	131 MBit
Ring Capacity	32	128	256
Ring Protocol	Token Ring	Slotted Ring	Register Insertion
Memory Capacity (max)	8K Bytes	128K Bytes	2 MBytes
Node Latency	2 ns	1 ns	125/660 ns
Node Separation (max)	50 ft.	50 ft.	0.7 km
Interrupt Fidelity	1	256	512,000
Byte Swapping	No	No	Yes
Data Filtering	Yes	Yes	Yes
Shadow Node	No	No	Yes
Bridge Node	No	No	Yes
Gateway Node	No	No	Yes
Monitor Node	No	No	Yes
Bus Interfaces	QBus	QBus, UNIBUS	All*
Message Parity Bits	1	1	9

* Being Implemented for all popular computer buses.

Current Design

Fiber Optic Link: Again, refer to Table 3 for a list of the major SCRAMNet Network design features. Of these, one of the most important is the use of a fiber optic transmission link.

The SCRAMNet Network design has utilized a fiber optic transmission approach as an integral part of the basic design--not as an optional add-on. It is the only transmission medium supported. This underlying design approach has permitted both the speed and simplicity of the system to be optimized to the point that would not have been possible otherwise. This was an important design philosophy and has significantly helped the network's performance.

The waveguide used is a 62.5/125 um specification, using a dual fiber link. In fact, the waveguide and connectors used are fully FDDI compatible, and thus will permit users to operate either FDDI or SCRAMNet Nodes over the same installed cables. This compatibility provides a migration path for FDDI users who find that their system cannot handle the requirements and need to upgrade to a shared-memory approach.

Although the 62.5/125 um cable is specified for SCRAMNet Network use, it will operate with a range of cable specifications. This includes the full range of FDDI specifications, both "allowed" and "preferred" cables. The particular cable can be optimized for the specific application environment and customers' needs. Using the 62.5/125 um cable, a maximum node separation of 700 meters is permitted within the allowed SCRAMNet Network attenuation/bandwidth budgets.

The transmitters used are dual, 820nm GaAs LEDs with ST connectors. Transition is made to FDDI connectors for connection at the computer-node level.

The receivers used are PIN diode units and include built-in transimpedence amplifiers. On-board connectors are again of the ST type, and transition is made to FDDI connectors at the computer-node level.

For applications where not all computers are powered-up for system operation, an optional fiber optic bypass switch is available. This switch is normally closed and will guarantee ring integrity when the computer is powered-down or when it chooses to eliminate itself from ring traffic. A bit in the node Control & Status Register (CSR) is used to allow the software control program to include/exclude itself from ring participation by operating the switch. This switch is recommended for some applications, but not needed for others. Thus, the optional feature.

The ring transmission methodology is a very novel and efficient protocol specifically tailored to shared-memory communications. By concentrating only on shared-memory communications with the design, the network has throughput and error recovery features far beyond those otherwise possible.

The data transmission rate is 131 MBit per second. Both fibers are used in the transmission scheme. The design budgets allow for a Bit Error Rate (BER) of less than 10^{-11}, but the protocol can easily tolerate a BER of 10^{-9}.

Data Communications: Communicating data over the SCRAMNet ring is both simple and fast. The software system architect needs only to configure the system-wide common data as a contiguous dataset, just as he would a FORTRAN COMMON, JOVIAL COMPOOL, or Ada Package. Then, this common dataset is included in each application program compilation unit, and its starting address is linked to begin at the first address of the physical shared-memory area. Once this is accomplished, data communications are both fast and automatic. Each time a common variable is updated (that is, one located in the shared-memory window), it is automatically transmitted to all other nodes on the network ring. No software is required for this process.

Refer to Figure 6 for a functional block diagram of this process. The CPU "writes" a dataword (32 bits) to a shared-memory memory location (that is, a memory location on the SCRAMNet card) by simply executing a high-level language assignment statement, such as ALTITUDE = CURRENT_ALT. The SCRAMNet memory "looks" to the CPU as all other memory on this CPU memory bus. However, when the variable ALTITUDE has been linked to the shared-memory area, this memory "write" by the CPU results in automatic and "user-transparent" activity by the SCRAMNet Network electronics. These electronics continuously monitor all "writes" to this shared-memory region, and each such "write" results in a single and automatic ring transmission. This ring transmission consists primarily of the memory word address and word contents. Circling the network ring, this 81-bit long message is received and re-transmitted by each node on the ring, and each node places the memory word contents in its SCRAMNet memory at the same relative physical address as the other nodes (at the relative memory address passed with memory word contents).

Simplicity and speed are the key features of this technology. Using this shared-memory scheme allows the application programmer to "view" the multi-computer system as one "virtual" multi-processor, where each CPU shares a common memory module. This simplified software architecture is the source of many of the strengths offered by shared-memory networking.

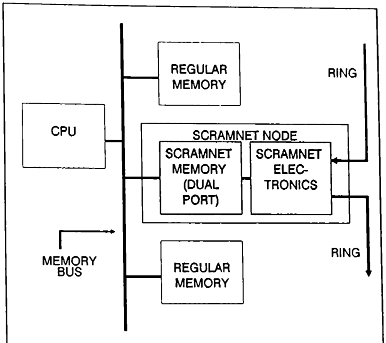


Figure 6
SCRAMNet Network Operation

Control Communications: In many real-time applications, "positive" real- time control over the distributed computers is a major requirement. This is particularly important where asynchronous events and external stimuli dictate this type of system operation. It is also a key requirement where synchronization is needed--e.g., in synchronizing models to clock frames.

Hardware interrupts are the key tool of the system architect when handling these system designs. Interrupts must be communicated among computers efficiently, fast, and "deterministically" (that is, with a small variability in the communication delay time).

To satisfy these requirements the SCRAMNet Network provides a wide range of options for controlling with interrupts over the network link. Interrupts are passed as data words--with the exact same speed and efficiency as data are.

Refer to Figure 7. This diagram shows the functional operation of an outgoing interrupt (that is, the action by one CPU to interrupt another on the network). A similar diagram can be drawn for the receiving-CPU side of this transaction.

To affect this transmission of an interrupt, the application programmer merely makes a high-level language statement in a manner analogous to communicating data--e.g., MISSILE_LAUNCH = GO. If the SCRAMNet Auxiliary Control RAM (ACR) bit has been previously set for the shared-memory word where MISSILE_LAUNCH is located, this "write" to this SCRAMNet physical memory location will result in an "Interrupt Bit (IB)" being sent over the SCRAMNet Network along with the memory word contents (D) and address (A).

The memory word contents, address, and interrupt bit are passed to all nodes on the network. In turn, any receiving node which has set its ACR to receive interrupts at this particular memory word location will be interrupted. Note that this is a very selective process--both the transmitting and receiving nodes must cooperate in this process. This provides the system designer with a very "positive" way to communicate interrupts and to control the distributed system operation.

As with data communications, speed and simplicity of communications are the key features of this shared-memory interrupt scheme. Interrupts pass in exactly the same way data do, at the same rates, and with the same deterministic performance.

Integer Reformatting: With the diversity of computer vendors, hardware data format compatibility has become a problem for the real-time designer. One aspect of this format problem is found with integer data representations. For example, a DEC processor may store

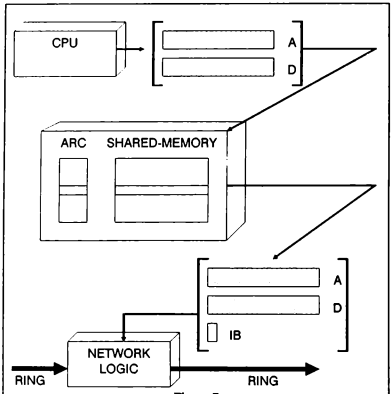


Figure 7
Outgoing Interrupts Functional Operation

integer data in a byte-ordering format which is different from a Motorola processor. Thus, communicating data between these processors is difficult and time-consuming if these conversions must be handled in software.

There are two major integer data formats describing how a processor associates the address of a byte with its significance in the data type in which it is contained; (1) Little Endian, and (2) Big Endian. Little Endian is the integer format where the least significant byte of data is stored at the least significant address. In the Big Endian integer format, the most significant byte of data is stored at the least significant address. DEC and Intel both manufacture processors and mini/micro systems which use the Little Endian integer format, where Motorola manufactures processors and systems using the Big Endian format.

The SCRAMNet Network provides a hardware reformatting solution to this problem. It is implemented by the application programmer by programming bits in the SCRAMNet interface, and it functions as shown in Figure 8.

The key points for the real-time architect are its speed and flexibility. It is programmable at each node. It is fast--completely transparent to the user and accomplishes the byte swapping in the normal memory cycle time.

Data Filtering: In many cyclical real-time systems, algorithms and tasks execute at fixed, clock-driven frame rates. This is typical of many aircraft simulation designs, where each model is assigned a particular rate-group. For example, 100 Hz, 50 Hz, 25 Hz, etc. This synchronous architecture scheme leads to a simple and efficient executive software structure, as well as a very deterministic system performance.

However, cyclical simulation systems such as these result in new model output data values being computed more frequently than needed in many instances. This stems from the basic nature of the fixed scheduler--models are put into frame rate-groups for the maximum execution rates ever needed by the models. Thus, when models are operating in their predominant "quiescent" environmental states, many model outputs change slowly. There may be several model iterations in a row where the model outputs are actually unchanged.

Without some method of detecting these invariant output data, these data are needlessly transmitted among the computers. These redundant transmissions consume network bandwidth and result in poorer utilization of system hardware and software resources.

To provide the real-time system architect with a solution to this problem, the SCRAMNet design incorporates the implementation of a programmable data filter.

The SCRAMNet Network is unique in this respect, because only those "writes" to the shared-memory area that produce a data value change are actually transmitted to the other nodes on the network. For example, if the shared-memory location containing CURRENT _ALTITUDE in a node's shared-memory contains the value "2000" and the CPU writes the value "2000" to that location, then no network traffic will be generated. However, if any other value is written to the CURRENT_ALTITUDE location, then the new value will be passed around the network to update the other shared-memory copies. This data filtering technique is a powerful technique for cyclical, real-time systems where not all data change on each system timing frame. The SCRAMNet Network is the only network that uses this technique.

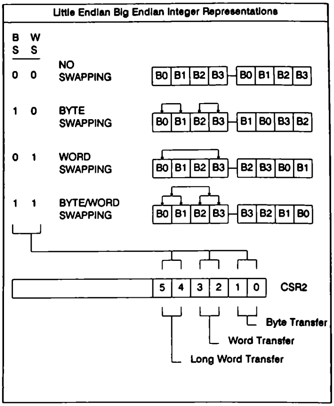


Figure 8
Programmable Byte Swapping

This technique has been shown to significantly increase the effective throughput of the network. Actual measurements on an aircraft flight simulation application have shown that this technique filtered out approximately 75% of the network traffic. This alone increased the "effective" bandwidth of the network by 400% for thi; particular application.

Figure 9 is a logical block diagram of the data filter operation. This filter is completely controllable by the application programmer, and it can be turned on or off to suit the application at hand.

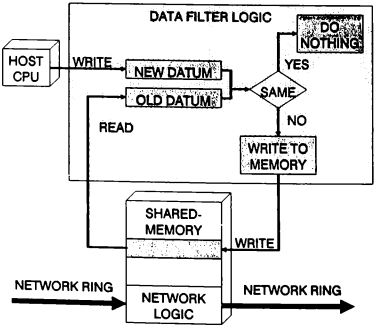


Figure 9
Data Filter Logical Operation

Shadow Nodes: Hardware redundancy, fault tolerance and graceful degradation are requirements found in many real-time systems. Although these are more prominent in life-threatening control situations, these are also found in some aircraft simulator applications.

A shared-memory network is a "natural" for these tough design requirements. With a shared-memory network all computers on the network can have all system state data all the time and with no additional network or CPU overhead. This is a very powerful system design concept, and it comes "free" with a shared-memory approach. Thus, error recovery and system reconfiguration can be both fast and efficient.

Coupled with this shared-memory network advantage, the SCRAMNet Network design provides the capabilities to configure up to four redundant networks in parallel--all operating with redundant data links and memories. The hardware is designed to accommodate this without any degradation of any of the networks, and no CPU overhead is involved.

Figure 10 shows a block diagram of four nodes in each CPU and with these configured in a shadow node operation.

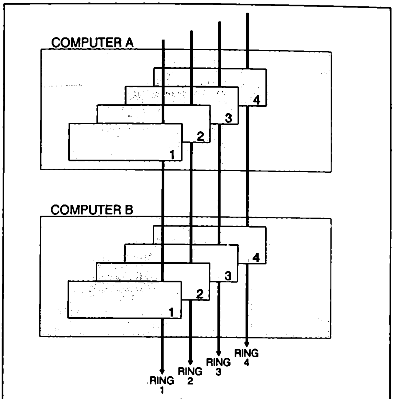


Figure 10
Shadow Nodes

Timing Considerations

Real-time systems are unique in the following respect: Outputs must be generated on-time, every time. This is the essence of the real-time domain and what makes it such a demanding engineering assignment.

Unlike many other high speed computing environments where "most of the time" is good enough or where "on the average" is good enough, most real-time applications demand correct outputs within rather stringent upper-bounds on the permissible time delays--and these must occur every time. For the inter-processor communications network, this demands both speed and "deterministic" performance. Usually deterministic means maintenance of very tight upper-bounds on data delays.

For an aircraft simulator, delays may be very critical for some parameters. Microseconds or a few milliseconds may make the difference.

The SCRAMNet Network protocol and architecture was specifically optimized for this deterministic situation, and its performance parameters reflect this design thrust. For example, if the value of the parameter CURRENT_ALTITUDE is updated in one computer on a ten node ring, this new value will be available in the memory of all other computers on the ring within 2.4 to 6.6 usec. This is the total application-to-application (A-T-A) communications time--that is, for instance, the time from when a FORTRAN program computes the parameter value and stores it in the variable CURRENT_ALTITUDE on one computer and when the value is available in the Ada application module on the farthest computer on the ring.

The particular ring design used is very deterministic. Messages are fixed in length; 81 bits long. No lengthy network access contention is involved. The ring access time is between 25 nsec and 120 nsec. The node latency time is between 240 nsec and 660 nsec per node, depending on whether the node is "active" at the time of data receipt.

Like data, interrupts are communicated over the ring with the same speed and simplicity of data. For the 10 node network example above, a master CPU could interrupt all nine other nodes within the same 2.4 to 6.6 usec elapsed time.

Much of the speed advantage of the shared-memory network comes from the fact that it is a "softwareless" communications link. Once the network node is set up, no additional software is required to make it operate as a real-time link. No real-time drivers are needed. Unlike a traditional message-passing protocol, there are no time-consuming software routines needed to pack, queue, transmit, de-queue, and unpack messages.

Additional Tools

To complement the SCRAMNet Network design, three general-purpose tools have been designed to support networking in real-time applications. All three are useful in aircraft simulator systems.

The concept of the Bridge Node is shown in Figure 11. This example shows the connection of two separate rings--one with a single aircraft simulator and one with two aircraft simulators.

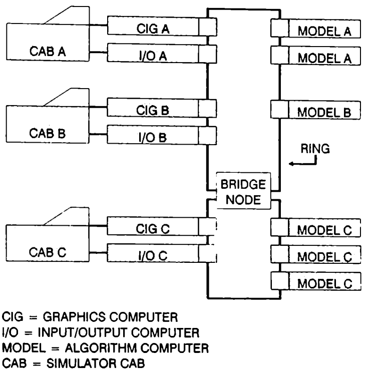


Figure 11
Bridge Node Example

With a shared-memory architecture, the connection of multiple simulators (e.g., for M-on-N applications) becomes a straightforward and efficient mechanization. The Bridge Node is, in essence, a high speed memory-to-memory mapping device. It can be programmed to map specific data locations on one ring to other data locations on the other ring. Its high speed operation and programmed selectivity allow it to function well in the multiple aircraft simulator environment.

A Gateway Node is also a useful tool in interconnecting to low speed, message-passing networks such as Ethernet and FDDI. A Gateway Node can be easily constructed using any existing node on the SCRAMNet Network. This is done by simply configuring one networked computer with both a SCRAMNet card and a direct memory address (DMA) type network card for the foreign network. The foreign network card is then programmed to DMA to and from the SCRAMNet memory area, and the Gateway is in essence accomplished. This implementation approach is both straightforward and efficient.

Third, a Monitor and Control (MAC) Node is a useful tool in some applications. The MAC node is a general-purpose node with additional capabilities for monitoring network traffic. A MAC node adds capability, but it is not required for normal network operation. This node can perform all the functions of a general-purpose node, but it has additional modes which can be set by the CSRs. These modes include the ability to constantly log time-tagged network traffic to a circular buffer, and the ability to trigger on an event from the network and capture a block of network traffic. The MAC node can be used to determine which node last wrote a data word in shared-memory, to generate an interrupt on various conditions, and to determine network loading and operating condition.

SUMMARY

Shared-memory networking, a "refreshing" new technology, offers the simulator system architect a powerful and efficient tool for interconnecting computer systems. This technology has been made feasible by advances in fiber optics, high speed electronics and electronic packaging technologies.

With a shared-memory network design, the system integrator accrues many benefits:

- Communications of data and control can take place at memory speeds--orders-of-magnitude improvement over message-passing networks.
- The overall system design can be conceptually "simple," easy to understand by all project personnel, easy to modify, and simple to test.
- Large savings can be made in both fiscal and schedule budgets, since system software and communications programming is greatly simplified.
- Deterministic performance can easily be achieved, due to the speed of communications and small variations in delay times.
- Real-time reconfiguration, real-time error recovery, fault tolerance and graceful degradation--all are easily implemented by-products of a shared-memory architecture.
- Computer hardware costs can be reduced by eliminating the CPU time needed for time-consuming communications software which is typical of message-passing systems.
- Systems can be integrated, expanded and modified quickly--thus saving both time and money for the system integrator.

Deciding whether a shared-memory network is the right solution for a project is a straightforward matter. Simply ask the question: Would this problem be best solved by a collection of computers all connected to a single shared-memory module? Pictorially this is shown in Figure 12. If this figure depicts the best "logical" configuration to solve the problem at hand, then a shared-memory network is the best "physical" approach. A shared-memory network makes a distributed computer system "look" to the system architect and software engineers as though it was configured as shown in Figure 12.

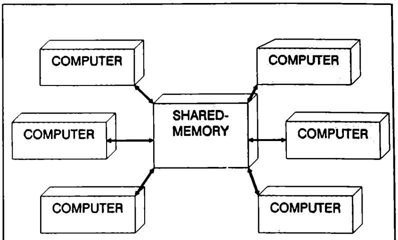


Figure 12
Is this the best "Logical" Solution?

SCRAMNet is a trademark of SYSTRAN Corp.

89-3300-CP

SYNCHRONIZATION AND TIME TAGGING IN DISTRIBUTED REAL TIME SIMULATION

Amnon Katz, Ph.D.*, Daniel M. Allen**, and Joseph S. Dickson+
McDonnell Douglas Helicopter Company

ABSTRACT

A system for synchronization and time tagging for distributed real time simulation is introduced. It is based on three "times": Time of Day (TOD), Mission Time (MT), and Dynamic Time (DT). All three are relative times reckoned in "ticks". TOD runs continuously and synchronously in all components of the simulation. It is used to insure synchronous transitions of all components to and from the RUN state. MT and DT are accumulated in RUN and frozen in other simulator state. MT is the mission clock representing time that the simulator is "flying" and the cumulative time from the (arbitrary) start of the simulated flight. Dynamic Time is a built in permanent tag for dynamic variables. DT is the MT at which the values are valid, as opposed to the time at which they were computed or used. Both MT and DT are tools for the analyst.

I INTRODUCTION

By definition, real time simulation runs as fast as the real world process it simulates. With proper calibration this holds true on the average. However, "in the small" the digital processes proceed by leaps and bounds and deviate considerably from the uniform flow of time. Time evolution is computed in steps. Typically time is divided into "frames". All of the processes that occur in parallel during the frame are computed during the frame, in sequence. Some of the frame time is used for computer housekeeping and communications. Some usually is surplus and remains idle.

The frame becomes the natural unit of time for simulation. It may be neither possible nor useful to measure an increment of time finer than a frame.

The situation is more complex when multiple processors, computing in parallel, are involved. Different machines may run different cycles. Some may be asynchronous, some may adjust their step time to the workload. Even when all run frames of equal duration, they are not directly comparable unless steps are taken to synchronize them. And even then, the correct time tag of any information may not be clear. Variables computed by processor X and used by processor Y may have been transmitted several times through a bus or local area network. The values used or recorded could be several frames old.

As more and more demand is made on simulation in the fields of training and testing of airmen, and of design and evaluation of airframes and avionics, the need to quantify the meaning of time tagging becomes more crucial. This is required in the assessment of simulation fidelity and for the correct interpretation of information gathered from simulation studies.

This paper outlines a system of timing for distributed real time simulation. Both hardware and software are involved. Three different "times" are invoked: Time of Day, Mission Time, and Dynamic Time. All three are relative times referred to an arbitrary starting point. All three are reckoned in "ticks". Time of Day is a continuously running count. Mission Time skips periods when the simulation is frozen. Dynamic Time addresses the distinction between the time a computation is made, and the time at which the result is valid.

In between them, the three time definitions provide the analyst with the tools to decide what takes place when, in simulation, and in the real world process being simulated.

II TIME OF DAY

Time of Day is defined by a common timing signal distributed to all processor boards. This signal may be the basic simulation frame count, or it may be a faster count whose frequency is an integral multiple of the frame count. In this case a new frame is started every nth count. Individual counts are available for timing finer than a frame.

Mission time is maintained as a tick count in a "register" or memory location that is available to all processes, real time as well as control functions, to read. In a distributed system, each processor or chassis maintains its own Time of Day count. This permits independent operation. It also ensures that time can be determined locally without undue delay and without an undue burden on intercomponent communications.

In the MDHC facility, Time of Day is a 59.94Hz signal derived from the image generation system and defining the basic frame of the out the window visual. The signal is distributed by coaxial cable to the many processors. Typically these are 68020 processor boards in VME chassis. The timing signal is wired into an unused line in the backplane, where every processor board picks it up. Each processor then maintains its own Time of Day count in an on board memory location. The processor can access its own time count over a local bus. At the same time the location is also available to external processes over the VME bus. In particular the System Control Station (SCS) has access over Ethernet to each VME bus and can read and write each processor's memory.

III SYNCHRONIZATION

The distribution by wire assures that the ticks counted by the different processors are synchronized to within nanoseconds (representing the differences in wire length). However, at this point, the Time of Day count maintained by different processors is different, since the processors are independent and start operation asynchronously.

*Manager, Advanced Development Office, Simulation System Group. Member AIAA.
**Advanced Dev. Office, Simulation System Group.
+System Integration Engineer, Sim. System Group.

Synchronizing the Time of Day counts of the different processors is a task to be accomplished as part of global system initialization by the System Control Station (SCS). The process consists of SCS reading its own Time of Day and writing the same value into the Time of Day register of another processor. Subsequently the SCS reads the TOD of the other processor to verify that a correct setting has been achieved.

SCS must achieve the reading of its own TOD and the writing to the remote processor within a single tick period. Similarly the verification must occur within a single tick period. Note, however, that the verification need not take place in the same period as the writing. Note also that TOD for different processors may be set in different periods. The process may proceed as follows:

SCS TOD	SCS ACTION
117	Set TOD for processor #1 to 117.
118	Verify that TOD for Processor #1 is 118.
119	Set TOD for processor #2 to 119.
120	Verify that TOD for Processor #2 is 120.
121	Set TOD for processor #3 to 121.
122	Verify that TOD for Processor #3 is 122.

The need of achieving each of the actions above within a single tick places an upper bound on the frequency of the timing signal. But this bound is not overly restrictive. 60Hz timing presents no difficulty.

IV MISSION TIME

One of the advantages of simulation over training or testing in flight is the ability to freeze the action. A difficult or otherwise significant situation may be "frozen" for discussion, consideration, or study. Sunsequently the simulation may be "unfrozen", and the flight continued from the point at which it was stopped.

Misson Time (MT) is a count of the TOD ticks that increments while the simulation is running, but remains constant when the simulation is frozen. More formally the definition is:

Mission Time is the count of TOD ticks that occur while the simulator is in the RUN mode.

Ticks that occur while the simulator is in any other mode - FREEZE, INITIALIZE, SHUTDOWN,... - are ignored.

Several obvious applications of mission time are:

o Timed maneuvers such as holding patterns and procedure turns need MT to complete correctly, if frozen.

o MT represents the cumulative time that has been flown and the simulator time that should be logged.

o Where environmental conditions, such as sun angle, or ambient temperature, are programmed to change in time, it is mission time that they should be tied to.

o Time histories of anything from variables of state to a complete tactical mission are most meaningful when stated in terms of MT.

V SYNCHRONIZED "FREEZE" AND "RUN"

It is fairly straight forward to count ticks when the RUN flag is true and ignore them when it's false. But a method is necessary to cause multiple processors to raise and lower their RUN flag on the same tick.

At this point we assume that the TOD synchronization described in section III has been carried out. Let us further assume that all processors are "frozen", i.e. their RUN flag is false. Our purpose is to cause all processors to raise their RUN flags in unison, i.e. on the same tick. It may not be possible to communicate with all processors within the same tick. On the MDHC control network, which does not support broadcast, it is definitely impossible. The way around this limitation is to lead the RUN command by several ticks. SCS does not issue an immediate RUN command. Rather it commands all processors to unfreeze on a named future tick. All processors have the command ahead of time. Once TOD reaches the value named, all processors raise the RUN flag together. This might proceed as follows:

		RUN FLAGS			
SCS TOD	SCS ACTION	PROC. #1	#2	#3	..
117	Instruct Proc. #1 to RUN at 122	F	F	F	...
118	Instruct Proc. #2 to RUN at 122	F	F	F	...
119	Instruct Proc. #3 to RUN at 122	F	F	F	...
.		.	.	.	
.		.	.	.	
121		F	F	F	...
122		T	T	T	...

A synchronized FREEZE is achieved the same way.

It is the timing of a synchronized RUN that is the main function of TOD. All simulation data are more meaningfully tagged by MT.

VI DYNAMIC TIME

An essential ingredient of any real time simulation is the integration in time of equations of motion. This proceeds in steps and uses the variables of state at the end of the last step (and possibly of previous steps) together with other variables (such as control inputs) to predict the values of the variables of state at the end of the step.

Values produced in this way are "valid" (i.e. are the approximation selected) for the precise time of the end of the integration step.

The computation normally takes place within the step and is finished before the end of the step. At that time the new values are usually put in memory to replace the earlier values. Yet these values are not, strictly speaking, valid all the time they reside in memory, nor even at the time they are first placed there. They are valid at the end of the step for which they are computed.

Even under the simplest circumstances, where a step, a frame, and a tick are all equal, an ambiguity exists. Values read from memory while MT has a certain value could be the values of the end of the last frame (if this frame's computation is not complete) or of the end of this frame (if it is).

Many situations can arise that are considerably more complicated. A frame could last several ticks. A step of a slow process may last several frames. An integration process may run asynchronously with the frames and ticks and to adapt its step to the workload. Any of these circumstances increase the ambiguity. Values that are available in memory over a number of ticks, are actually valid for a specific tick.

The situation is even further complicated in a distributed system with multiple processors. Variables may be transmitted from processor to processor. Each transmission may occur in a different frame. In this way values may reside in one processor's memory long after the step in which another processor originally computed them, and long past their instant of validity.

It is obviously desirable to permanently tag dynamic variables by the point in time for which they are valid. This accomplished by Dynamic Time (DT) defined as follows:

Dynamic Time is a dynamic variable whose time derivative is unity.

We append a variable T to any set of variables that are integrated together along with an equation of motion

$$dT/dt = 1,$$

and an initial condition

$$T_{init} = MT.$$

T becomes the Dynamic Time for that set of variables. The DT is kept in memory with the other variables and transmitted with them to other processors. In this way the precise moment of validity is always available. This device automatically takes care of the complexity of long integration steps and variable integration steps.

The equations above assume that ticks are the unit of time used. If this is not so, a conversion of units is required.

The DT so defined keeps pace with Mission Time. This is so because, by definition, integration of equations of motion is suspended in all but the RUN mode. Dynamic Time thus provides a permanent tagging of dynamic variables by the appropriate value of Mission Time.

VII DISCUSSION

Mission Time (MT) and Dynamic Time (DT) are both tools for the analyst. By reading dynamic variables together with their DT and also reading MT at the same time, the analyst secures a complete picture. DT is the time at which the values obtained were valid. MT is the time at which they were available and presumably being used. Both methods of tagging convey meaningful information. The spread between them measures the delays inherent in the system.

TOD is merely a tool necessary to insure synchronized changes in the RUN flag over the distributed system. This in turn is required for correctly accumulating both Mission Time and Dynamic Time.

The current trend in simulation is toward parallel processing on a large scale. This compounds the difficulties of correct and consistent time tagging. Yet, as the demands on simulation increase, so does the need for precise and rigorous timekeeping. Management of time along the lines suggested here satisfies that need.

89-3301-CP

FOUNDATIONS FOR TACTICAL TRAINING: A CHALLENGE TO INDUSTRY

Dennis M. Miller*

CAE-Link Flight Simulation Division
Binghamton, New York

Abstract

Training crews to tactically employ a weapon system requires heavy emphasis on cognitive learning. The quality and complexity of the cues received from the environment is crucial to the crew members' ability to make quality judgments about the tactical situation. Simply stated, environmental events must be consistent with what is known of the characteristics and the tactical behavior of both enemy and friendly forces; tactical training requires highly realistic combat environments.

The level of realism required to support tactics training can only be achieved when both *the appearance* and *behavioral* aspects of the combat environment are properly represented as they are perceived by the crew. The ordered nature of environmental cues is a manifestation of underlying rules that govern behavior, including physical laws, national policy, military doctrine, strategies, and tactics. The simulations of Orders of Battle (OOB's) and $C^{2/3}$ processes and structures are crucial elements that provide the appearance and behavioral qualities required for tactical environment realism.

This paper examines the mission-ready (MR) unit's most basic responsibilities and reveals requirements for the development of viable tactical training environments. It also presents a methodology for identifying lower-level requirements for simulating tactical environments for effective training in decision-making.

The Challenge

A vital need exists in today's mission-ready (MR) units for realistic tactical training, specifically in the area of proper employment of sophisticated weapon systems. Effective and efficient weapon system employment is a critical factor that can determine the outcome of modern-day military conflicts.

Peacetime training is the single means by which MR units and their personnel gain knowledge of, and maintain proficiency in, the combat skills necessary to successfully complete wartime missions. Efforts to improve training in weapon system employment must fully consider each unit's responsibilities, as defined by their respective military command structures, and the environment in which they are required to perform their mission. The training need is centered on Mission Ready Crews, i.e., crews who are experienced and proficient in the operation of mission equipment and certified for combat after passing various mission qualification evaluations conducted in both training devices and actual weapon systems.

In peacetime the training environment is subject to many constraints, which exist generally in areas of policy and regulation, available funds, and technology. The lack of affordable technological solutions to tactical training issues has necessitated the enactment and enforcement of regulations which ensure acceptable margins of safety and at the same time minimize obtrusions into the peace and tranquillity of civilian populations. Consequently, for example, aircrews cannot gain experience in visually acquiring and maneuvering their aircraft in direct response to surface-to-air missiles (SAM's) while the missiles are in mid-flight or dodging flak patterns and tracers. The only way to get this training now is to use real missiles and AAA, which by anyone's standards is not safe.

Noise abatement procedures restrict many tactical operations and maneuvers to special-use ranges and airspace. Thus, for example, practice of low-altitude tactical approaches and departures to and from airfields seldom occurs. In some areas where U.S. forces are stationed, unusually high minimum flight altitudes (800 - 1000 ft AGL) are levied against low-altitude operations for noise abatement. MR pilots training under these types of constrictions cannot become proficient at low-level tactical maneuvering and can expect virtually no opportunity to experiment with and test variations on classical low-level tactics.

Exercises such as Red Flag, Checkered Flag, Green Flag, Reforger/Crested Cap, Cold Fire, etc., are conducted each year at great expense in pursuit of training that will increase combat readiness. As valuable as these exercises are in contributing to combat readiness, they are still subject to severe and necessary constraints. Few MR crew members ever get to participate in large exercises like Red Flag, and those that do participate infrequently, due mainly to lack of funds and availability of properly instrumented ranges. A perpetual backlog of crews needing to participate continually far exceeds the throughput capability of available facilities, severely limiting the ability of mission-ready units to attain their full combat potential.

Any effort to advance the quality of training provided to MR crews must implement approaches directed at accomplishing specific mission-oriented learning objectives. This requires a systematic process that allows for creativity and intuition on the part of industry, customers, and users. It also requires personnel who have a working knowledge of the available technologies, and are thoroughly familiar with the needs and the problems associated with military training. **The challenge to industry is to properly apply existing and emerging technologies in such ways as to alleviate present constraints on performing realistic training during peacetime and thereby increase the combat readiness of MR units.**

This paper examines the mission-ready unit's most basic responsibilities for fundamental training requirements that form the essential foundation for the development of a

* Member AIAA

tactical training environment. Emphasis in this paper is on Air Force training (due to the writer's experience), but statements about training are equally relevant for all MR crews, regardless of service branch or weapon system (M-1 Tank, F-16, C-17, V-22, B-1B, submarine, aircraft carrier, etc.).

Driving Forces

The search for specific ways to improve training, within and of operational units, must begin with a clear understanding of frequently used military terms. "Mission" and "Weapon System" are two such terms, which are explicitly defined by the Joint Chiefs of Staff.

Mission — The task, together with the purpose which clearly indicates the action to be taken and the reason therefor.

In common usage, especially when applied to lower military units, a duty assigned to an individual or unit; a task.

The dispatching of one or more aircraft to accomplish one particular task.

JCS Pub 1[1]

Weapon System — A delivery vehicle and weapon combination including all related equipment, material, services and personnel required so that the system becomes self-sufficient in its intended operational environment

JCS Pub 1[1]

Notice that a "Mission" is a task and applies equally to individuals as well as it does to units as a whole, from the squadron to the wing or even to an entire service branch. The missions of the Air Force, as defined and described by AFM 1-1, Basic Aerospace Doctrine of the United States Air Force,[2] are shown in Fig. 1, as an example.

MISSIONS OF THE USAF

o Strategic Aerospace Offense

o Strategic Aerospace Defense

o Counter Air
—Offensive Counter Air (OCA)
—Defensive Counter Air (DCA)
—Suppression of Energy Air Defense (SEAD)

o Air Interdiction (AI)

o Close Air Support (CAS)

o Special Operations

o Airlift

o Aerospace Surveillance and Reconnaissance

o Aerospace Maritime Operations

Fig. 1 Missions of the USAF.

The mission is the source of all operational requirements. It is the driving force behind all efforts that the combat-mission-ready unit expends in training to meet its operational commitment. The combat unit exists for the purpose of accomplishing its assigned mission, be it Close Air Support, Air Interdiction, or any combination of assigned missions. The unit's mission is complete when each unit member successfully completes his individually assigned missions (e.g., destroy a command post, attack an airfield, or impede the advancement of a tank regiment).

The nature of the tasks involved in executing military missions invariably requires the use of sophisticated weapon systems. The definition of "weapon system" reveals that, in addition to the usual machine ingredients of hardware, software, and firmware, personnel who are required to operate the equipment are also included. When one discusses, for example, the F-4 Weapon System, by definition the crew members (Pilot and WSO) are included in that definition and are integral and indispensable components, without which the F-4 ceases to qualify as a weapon system.

Thus, an axiom about the MR crew can be stated:

The responsibility of every mission ready crew member in an MR unit is to accomplish the mission through proper employment of the assigned weapon system.

Needs & Tasking

It is imperative that we listen to what crews say about their training needs. The statements heard most frequently are "We want to train like we fight", or "We want mission oriented training", or "We want to do mission rehearsals." The Chief of Staff of the Air Force, General Larry Welch, confirmed this need at the highest levels of the military when he stated, " *The key to unlocking the combat capability inherent in quality people and quality equipment is training the way we will fight.*"[3] Schultz, Owens and Harris defined 'mission-oriented training' as, " *.......training that encompasses 'intra' and 'inter' aircraft crew coordination and training in special tactics and missions.*"[4] Clearly, MR crews need a training environment that allows them to learn, develop and apply tactics. To determine how the training industry can contribute to fulfilling the military's need to *train like they fight*, we must return to the 'mission' and identify the requirements that reside therein.

The Air Force specifies each unit's wartime mission in a Designed Operating Capability (DOC) Statement.[5] In the Mission Tasking Narrative section of this document a description of the unit's mission(s) is given.

A closer look at the example narrative in Fig. 2 reveals that a unit can be tasked to perform more than one type of mission (role). The circumstances under which those missions are expected to be carried out are also specified; in this case the condition is General War.

The mission narrative is the route source for a top-level description of what the unit must do to train for its missions, based on what it must do to execute the mission in actual combat conditions. Considering the full spectrum of combat skills required to fight in a war, and the constraints upon training those skills in peacetime, it is clear that there are serious gaps in the training provided to MR crews. Clearly mission-ready crews and units simply do not have the resources needed in preparing for war in peacetime. The available instructional tools and practice

MISSION TASKING NARRATIVE

Provide Tactical Air Support for the Contingencies of General War.

Primary Capability: Employ Conventional Munitions, including AGM-65 Maverick against surface targets in the following roles:

- Air Interdiction
- Airfield Attack
- Defense Suppression
- Close Air Support

Secondary Capability: Employ Nuclear Weapons Against Surface Targets

Conduct Defensive Counter Air (Air Defense) Operation

Conduct Offensive Counter Air (Air Superiority) Operations

Fig 2. Mission tasking narrative.

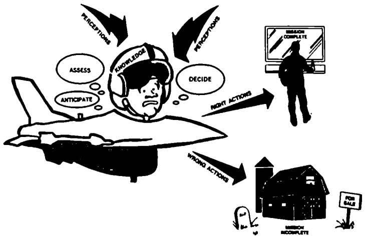
environments cannot accommodate the unrestrained training essential to developing the full potential of a fighting unit. For the operational unit to maximize its combat potential, the training performed in peacetime must coincide very closely with wartime tasks and conditions. To do this the training environment must contain elements that are present in the combat environment which are critical in learning how to tactically employ a weapon system.

Decision Skills

In addition to understanding the combat unit and its missions, an analysis of the various skills on the part of mission-ready crews responsible for accomplishing the mission is required. Psychomotor skills such as those required for equipment operation and procedure execution are important and necessary, but in mission-ready units these skills are already well established, and are practiced only to the extent that required proficiency is maintained. Other skills are required, and these skills are cognitive in nature. As Schultz, et al. have stated, "*Superior pilots will ... be those with the greatest system knowledge, decision making skills, and ability to maximize the utilization of system resources to achieve mission objectives.*"[4] The need for decision skills is common to all mission-ready units, but training decision skills at the operational unit level is the type of training least supported by existing programs and devices.

When performing an assigned mission, combat crews are required to make timely and correct decisions based upon their perception of the situation. Their survival and the survival of others, and ultimately the overall success of the mission, are dependent upon the quality of these decisions. Any training environment, to be useful to operational units, must support training that improves decision-making skills. Combat is characterized by rapid changes in conditions and by stresses and pressures which strongly influence crew performance. Crew members must constantly and accurately assess the current tactical situation, and at the same time anticipate conditions likely to impact them in the future, all under what is frequently perceived as time compression.

The decision process is composed of a number of successive steps including perception, assessment, anticipation, and decision, resulting ultimately in specific crew actions. Each step of this process is combined with the inherent knowledge of each individual to ultimately arrive at individual actions. Cues occurring in the present are perceived and assessed not only for their meaning and impact as to the current state of affairs, but also for what they might indicate about an anticipated time, even though that time may only be an instant away. Decisions are often made in anticipation of predicted circumstances. In other words, in combat, **tactical decisions are often based on how the future is perceived.**



The decision process requires exposure to patterns of complex and dynamic information reflecting sequences of events occurring rationally in the environment. In a tactical environment the ordered nature of this information reflects a set of underlying rules that govern behavior. These include physical laws of nature, national policy, military doctrine, strategies, and tactics. Except for the physical laws, these rules govern the performance of tactical elements through systems of command, control, and communication ($C^2/^3$). These concepts are summarized in the definitions of Decision, Doctrine, Weapon System Employment Concept and Tactics in Fig. 3.

DEFINITIONS

Decision: A judgment.

ref. Webster's New World Dictionary[6]

Selecting a course of action in the presence of uncertainty as to the immediate situation or in the absence of a predetermined procedure or doctrine.

ref. "Visual Cue Requirements in Imaging Displays"[7]

Doctrine: Fundamental principles by which the military forces or elements thereof guide their actions in support of national objectives. It is authoritative but requires judgment in application.

ref. JCS Pub 1[1]

Weapon System Employment Concept: A description in broad terms, based on established outline characteristics, of the application of a particular equipment or weapon system within the framework of tactical concept and future doctrines.

ref. JCS Pub 1[1]

Tactics: Employment of units in combat. The ordered arrangement and maneuver of units in relation to each other and/or to the enemy in order to utilize their full potentialities.

ref. JCS Pub 1[1]

Fig. 3 Definitions.

The ability to reject unacceptable alternative courses of action through cognitive or reasoning processes without having to try out every possibility represents the highest level of tactical skill. The successful selection from among multiple possible courses of action requires judgment, which in turn, requires extensive training. *Decisions are judgments* made in the presence of uncertainty. *Doctrines are fundamental principles that require judgment in application.* Decision skills, thus, are requisite to the successful application of doctrine in war.

The substance of the mission-ready crew's decisions, to a large extent, has to do with the employment methods of a particular weapon system. Methods of employing a weapon system are composites of various tactics either selected or developed for the anticipated situation that, in turn, support tactical doctrine. Aircrews confront the dynamics and uncertainty of combat through careful planning, development, and selection of tactics.

Training in tactics is a largely cognitive process. It requires instruction and practice with information representing complex, dynamic tactical environments and events. It requires the encouragement of personal initiative and innovative thinking on the part of the MR crew members to optimize the application of tactics in representative missions, tasks, and mission conditions. The training environment must include the information needed to ensure that the decisions reached are valid and transferable to real combat environments. In addition, frequent practice is required to maintain proficiency in decision-making skills.

The conditions of the tactical arena are so dynamic that it is impossible to dictate specific tactics for all possible situations.

> *"There is no approved solution to any tactical situation."*
>
> **Gen. George S. Patton Jr.**[8]

It follows, then, that the employment of a weapon system in a tactical environment is truly an art which cannot be taught or learned by a strict procedural process.

Historically, aircrews have the lowest chance of survival during the first seven to ten combat missions. After this initial exposure and the inevitable learning it fosters, the probability of survival increases greatly. The ability to practice representative missions in simulated environments containing information which is combat relevant, dynamic and uncertain is essential in replacing the learning otherwise required in the first few combat operations. The instructional system must stress exposure to information reflecting combat conditions and dynamics to adequately support the learning of tactics and the honing of the decision skills necessary to the effective employment of a weapon system.

The learning of tactics is stressed by the Air Force, as evidenced in the multi-command manual 51-50 Volume I.[9] Fig. 4 summarizes many of the requirements for tactical training contained within that document. Instruction in tactics both on the ground and in flight is required for all missions assigned to the Tactical Air Forces. Furthermore, "Exercises should be designed to provide *valid tactical training.*"[9]

Training MR crews in the tactical employment of a weapon system lies largely within the cognitive realm of behavior modification. The quality of the information received from the environment is crucial to the crew members' ability to make quality judgments about the tactical situation. Simply stated, environmental events must make sense in light of known force behavior associated with enemy and friendly force tactics. Thus tactical training requires exposure to and practice in many highly realistic tactical settings, each containing realistic amounts and kinds of information. Forgoing realism in the training environment is a dangerous choice that can prove fatal to both crew and mission in times of military conflict.

51-50 Volume I

Chapter 3, Section A

"Category I: Mission essential training that focuses on tactical subjects including . . . tactics . "

"Electronic Warfare: . . . enemy threat systems and tactics."

"Air-to-Surface . . . tactical employment concepts . . ."

"Air Combat Training: . . . tactical formation . . . tactics . . . air-to-air weapons employment considerations . . . . . enemy tactics."

" . . . emphasize . . . complete the mission."

"Intercept: . . . command and control . . . rules of engagement . . . intercept tactics . . . "

"Reconnaissance: . . . tactics . . ."

"Specialized training . . . tactics . . ."

Chapter 4, Section B

"Training Concepts: . . . use realistic training scenarios and profiles which simulate the crews' wartime tasks. Avoid stereotyped tactics . . . . Exercises should be designed to provide valid tactical training."

Fig. 4 Requirements for tactical training.

Realism

Providing a realistic environment suitable for learning and developing tactics requires an understanding of the information contained in the real environment, and of its relationships to crew performance and learning. In simulated environments such as those found in weapon system trainers (WST's), the emphasis has been placed on the faithful representation of the physical elements of the environment. For example, in visual scenes, realism has been translated to mean more highly featured terrain, "real" looking trees, and skin rivets or paint markings on aircraft. In the electronic threat environment, realism has been understood to mean precise signal simulation or generation. These efforts and others like them contribute to realism, but only to a limited extent, and by themselves can never produce an environment suitable for the development of the higher-order decision processes required to execute tactics.

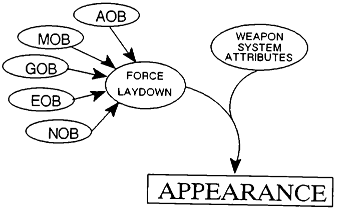
There are two essential ingredients of realism; *appearance and behavior.* Appearance describes the physical characteristics of the environment and its elements. Of the five human senses, sight must be given special consideration because it is through sight that crew members receive the vast majority of information about their environment. Visual cues are largely responsible for how crew members perceive the current tactical situation. Appendix A lists many of the objects seen in the combat environment that have significant impact on tactical decision-making.

In contrast, behavior describes how the environment and its elements act and react. Behavioral attributes range from those that are specific to a weapon system, due to its design, to the maneuvers of combat units which act in accordance with doctrines, strategies, tactics, and orders, in pursuit of mission objectives. The behavioral aspects of the combat environment are crucial in achieving the level of realism needed to support tactical learning objectives.

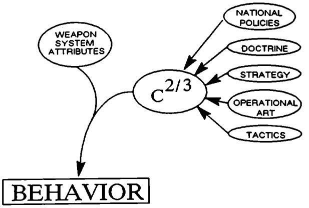
Appearance + Behavior = Realism

Requirements for simulated training environments are derived from the definition of tactics, within the need for realism in training. By matching the expression of realism against the definition of tactics some striking relationships become apparent that easily transform into requirements for simulated environments. The first segment of the definition of tactics in Fig. 5, ". . . *ordered arrangement of ... units in relation to each other and/or the enemy*", concerns the composition and placement of forces on the battlefield. This is a major part of the appearance aspect of realism, and is commonly referred to as force laydown. Force laydowns are specifically defined in official Orders of Battle (OOB's). An Order of Battle is "The identification, strength, command structure, and disposition of the personnel, units and equipment of any military force."[1] OOB's are grouped into five major categories: Air Order of Battle (AOB), Missile Order of Battle (MOB), Ground Order of Battle (GOB), Electronic Order of Battle (EOB), and Naval Order of Battle (NOB). These **OOB's must be simulated** in order to provide the appearance aspects of realism — that is, to provide essential information about the appearance of the battlefield, its components and their arrangement.

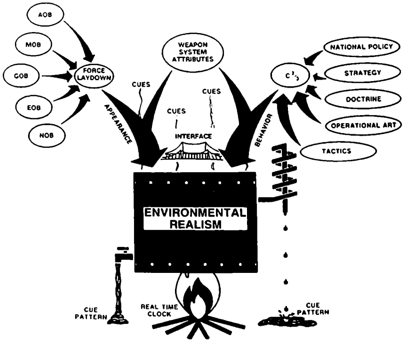
The degree to which OOB's are detailed and combined within a specific simulation is dependent upon the type of mission (Close Air Support, Strategic Defense, etc.) and the location in the world where the mission will take place, and on the training objectives to be addressed. Additionally, some attributes of the weapon systems contribute to the appearance aspect of the battlefield (shape, size, electromagnetic emissions, color, etc.). Each of these appearance attributes must be carefully examined for its impact on the learning of sound tactics and their execution, as well as, the improvement of decision skills for MR crews.



The second segment of the definition of tactics, ". . . *maneuver of units in relation to each other and/or the enemy* . . .", is a behavioral attribute. It is equivalent to real-world battle management when considering U.S. forces, or troop control when considering Soviet forces. Other political-military systems may have different terms and definitions for the maneuver of units, but the effects on the environment will always be behavioral in nature. The rules that govern the behavior of military units are deeply rooted within the intertwining of national policies, strategy, doctrine, operation art (Soviet term), and Tactics. The mechanism by which these principles and rules are conveyed and managed is through a systematic process of command, control, and communication ($C^{2/3}$). Therefore, it follows that, in order to replicate the behavior of military units in a simulated environment, **the $C^{2/3}$ aspects of the warring states must be modeled**, as well the rule bases that implement and execute national policy, strategy, doctrine, operational art, and tactics. There is no training environment available today, either real or simulated, that adequately represents this aspect of the combat environment.



$C^{2/3}$ must be properly modeled and interfaced with the OOB's, to ensure that the military forces represented will maneuver in a doctrinally correct and coordinated fashion, such that, realistic cue patterns and trends are formed. If $C^{2/3}$ is not modeled at all, no usable patterns or trends will occur, eliminating tactically significant behavioral patterns. Efforts on the part of the MR crew member to assess, anticipate, and decide will not be supported, making the learning of tactics impossible. If $C^{2/3}$ is not accurately modeled, the patterns and trends formed will evoke crew decisions that most probably will be inappropriate in similar real-world situations. Thus, training will be invalid or negative with respect to subsequent real-world performances. Schultz, et al. indirectly underscored the importance of behavior in simulation by their criticism of current approaches to training when they stated, "*Sequences of specific actions* (crew actions) *will depend on the events that arise as the operational scenario unfolds. Since the tasks can seldom be described as fixed, deterministic sequences, operators cannot be adequately trained by drill or fixed scenarios.*"[4] This is to say that training devices designed to operate only with scripted missions are of limited use to operational units whose greatest need is in learning tactics and tactical weapon system employment concepts in tactical environments which both appear and behave realistically.



REALISM AND TACTICS

REALISM = APPEARANCE + BEHAVIOR

TACTICS: The ordered arrangement and maneuver of units in relation to each other and/or the enemy in order to utilize their full potentialities.

TACTICS DEFINITION	REAL WORLD	SIMULATED ASPECT	SIMULATION REQUIREMENT
". . . ordered arrangement . . . of units in relation to each other and/or the enemy."	Force Laydown	APPEARANCE	Model Orders of Battle (OOB's) AOB MOB GOB EOB NOB Weapon System Attributes
". . . maneuver of units in relation to each other and/or the enemy . . ."	Battle Management (U.S.) Troop Control (Soviet)	BEHAVIOR	Model $C^{2/3}$ in accordance with national policies, doctrines, strategies, operational art, and tactics. Weapon System Attributes
". . . to utilize their full potentialities."	Successful completion of the mission.	Appropriate level of Realism APPEARANCE + BEHAVIOR	Interface OOB's and $C^{2/3}$. Realistic environmental cues that are Tactically Significant.

Fig 5. Realism and tactics

The final segment of the tactics definition , "*. . . to utilize their full potentialities*", is fulfilled when the crews have maximized the utilization of the weapon system resources and achieved mission objectives. **Successful completion of the mission is the culmination of training the way they intend to fight.** A simulated environment that substantially and accurately embodies the appearance and behavioral aspects of realism is essential in satisfying the environmental requirements necessary for tactical training.

The ability of MR crews to employ weapon systems in a tactically sound manner is fundamental to the successful completion of the mission for operational units. Practicing tactics, especially while ingressing and egressing the target and even delaying within the target area as mission requirements may dictate, results in better-trained crews. Similarly, being able to test, modify, and refine a tactical approach when planning a mission affords crews the opportunity to learn beforehand which tactic(s) holds the best promise for mission success. The opportunity to optimize tactics prior to mission execution greatly enhances crew preparation. Better-trained and better-prepared crews increase combat readiness and reduce combat losses, which in turn increase the probability of reaching the bottom line: **completing the mission**. A training system that supports the learning, instruction, and optimization of tactics will make a significant contribution to combat readiness and actively function as a force multiplier.

A Methodology

Achieving realistic tactical training requires the development of an instructional system directed at specific training objectives relating to the skills needed in tactical operations, coordination, and decision-making. Such a system requires the implementation of a systematic process whereby crew skills and significant attributes of the operational environment are identified, described, and related to the psychological process by which they are exercised and learned. The process begins with the generation of mission descriptions, detailed task lists, and the identification of mission-critical tasks and environmental conditions. Furthermore, a sorting and prioritizing scheme for tasks and environmental conditions must be defined. The sorting and prioritizing process should clearly establish each task's level of importance relative to successful completion of each mission. Additionally, analyses that define task difficulty, learning objectives, and training objectives must be performed. Finally, these mission level analyses should be recorded in a Mission Description/ Analysis document.

In the same way that Military Doctrine and Mission Tasking Narratives are driving forces for MR units, so too must they be driving forces for learning and training analyses. Each mission assigned to the MR unit in question should be described in narrative form so that design engineers and programmers, who are unfamiliar with the mission, will gain a sound understanding of how specific military operations are conducted. Mission Tasking Narratives must be critically analyzed until major tasks with subtasks are identified and collated to form an initial mission level hierarchical list. Fig. 6 illustrates a very high level breakdown of the example Mission Tasking Narrative presented earlier in this discussion.

OPERATIONAL / TRAINING OBJECTIVES

TASK	CONDITIONS
1.0 Accomplish the assigned mission.	Through proper employment of the weapon system.
1.1 Provide Tactical Air Support.	For contingencies of general war.
1.1.1 Employ conventional munitions.	In the conditions of a general war environment:
1.1.1.1	– against surface targets
1.1.1.2	– in the Air Interdiction role
1.1.1.3	– in the Airfield Attack role
1.1.1.4	– in the Defense Suppression role
1.1.1.5	– in the Close Air Support role
1.2 Employ the AGM-65 Maverick.	In the condition of a general war environment:
1.2.1	– against surface targets
1.2.2	– in the Air Interdiction role
1.2.3	– in the Airfield Attack role
1.2.4	– in the Defense Suppression role
1.2.5	– in the Close Air Support role
1.3 Employ nuclear weapons.	In the condition of a general war environment:
1.3.1	– against surface targets
1.4 Conduct Defensive Counter Air (Air Defense) Operations.	In the condition of a general war environment:
1.5 Conduct Offensive Counter Air (Air Superiority) Operations.	In the condition of a general war environment:

Fig. 6 Operational/training objectives

The initial task list is then expanded as required until the analyst is confident that all tasks germane to the mission, irrespective of any weapon system, are identified. These tasks, in turn, should then be prioritized by use of two filters, "*mission uniqueness*" and "*mission criticality*":

Mission-Unique Tasks — *Operational tasks peculiar to a mission type.*

Mission-Critical Tasks — *Operational tasks that are required to be performed correctly in order that the objectives of the particular type of mission may be fulfilled.*

A task may qualify as both "mission-unique" and "mission-critical." Where this is the case, special scrutiny is warranted by the training analyst to ensure that adequate capability is designed into the device to support the learning of these tasks.

The mission-unique sort will identify tasks that are peculiar to a selected mission. Early identification of mission-unique tasks is essential, since unusual device requirements may emerge as necessities in the facilitation of learning these tasks. An example of a mission-unique task might be: *Launch authentication procedure for a nuclear strike mission.* The list of tasks meeting mission-unique criteria will generally be small but may have a significant impact upon design.

In like manner, the mission-critical sort identifies tasks that must be performed correctly in order that the mission objectives will be accomplished. An example of a mission-critical task would be: *Making the time-over-target (TOT) by -/+ 30 seconds.* Mission critical tasks must be the focus for training requirements and device design. Engineering efforts must ensure that the robustness of the design is sufficient to support the instruction and learning of these tasks as well as all their associated contingencies.

Similarly, as with the crew tasks, environmental mission conditions and mission events must be analyzed. A useful criterion to aid the analyst in this effort is *tactical significance:*

Tactical Significance — *The degree of importance an environmental event or condition has, in a given setting, upon decisions made with regard to tactical situations, hence upon future courses of crew member actions.*

An example of a *tactically significant condition* for an A-10 on a Close Air Support mission would be: *Target weather equal to 1,000 ft overcast ceiling and 3 miles visibility.* An example of a tactically significant event for an F-111 on an Interdiction mission would be: *A puff of smoke on the ground at 10:00 o'clock position followed by a smoke trail and no RHAW indications.*

Conditions and events that are tactically significant are dependent upon the specific mission(s) and the weapon system(s) assigned to each combat unit. The same cue in the environment can have different values of tactical significance due to the nature of the mission and weapon system performing the mission. For example, a tank maneuvering along a line of trees would have great tactical significance to Apache or Scout helicopter crews, but would have very little tactical significance to B-52 or B-1B aircrews at an altitude of 35,000 feet. **Tactically significant conditions and events must be determined by operations specialists who are close to the unit's mission and concept of operation or by operational crews themselves. Both the system developer and its potential user must have a dedicated team to address this issue for each program.** The conditions and events of the mission should be combined with each task to form a single list of mission level operational/training objectives.

Military doctrine and tactics are the governing factors influencing mission level analyses. Doctrine and tactics change very slowly over time; thus, changes to the mission analysis findings will be relatively infrequent. However, periodic review and update of these analyses are needed, primarily due to influences technological advances have on doctrine and tactics. A single updated list of operational/training objectives is necessary in ensuring that MR crews continue to have the capability to *train like they fight.*

From the list of mission level objectives and subsequent analytical processes an initial list of device requirements can be generated by operations and training analysts. These device requirements will provide an early input to engineers whereby they get a first look at the magnitude and breadth of the engineering tasks that lie ahead (Fig. 7).

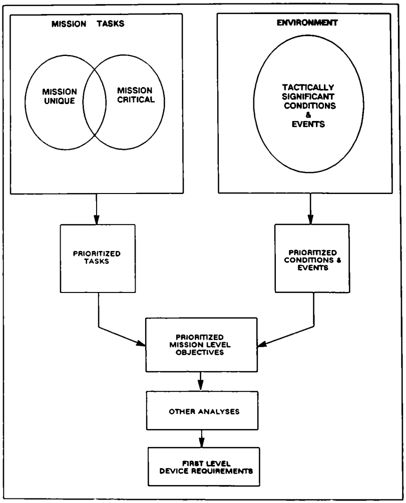

MISSION TASKS
MISSION UNIQUE
MISSION CRITICAL
ENVIRONMENT
TACTICALLY SIGNIFICANT CONDITIONS & EVENTS
PRIORITIZED TASKS
PRIORITIZED CONDITIONS & EVENTS
PRIORITIZED MISSION LEVEL OBJECTIVES
OTHER ANALYSES
FIRST LEVEL DEVICE REQUIREMENTS

Fig. 7 Mission description/training analysis document

The analysis should concentrate on the specific weapon system to be simulated as well as the specifics of the environment in which it is expected to operate. The processes of this phase are much the same as during the mission analysis phase, but are conducted at a lower level (Fig. 8). This phase is designed to reveal more specific details about the training requirements, hence more detail about device requirements. The task list generated during mission analysis is expanded to include all tasks required of MR crews to properly operate and employ their weapon system. The list should include **decision tasks** as well as psychomotor tasks. An analysis of the decision tasks will reveal, primarily but not exclusively, characteristics of the environment external to the weapon system (tracers, blowing dust, smoke trails, threat behavior, flight formations, team interactions, etc.). Exceptions to this statement are weapon system malfunctions which occur internal to the weapon system. Complementary to decision tasks, analyzing psychomotor tasks will reveal characteristics about the environment inside the weapon system (operational switches, gauges, lights, handles, etc.).

The prioritization process at this level is aided by four additional task filters: Weapon System Unique, Weapon System Mission Unique, Weapon System Critical, and Weapon System Mission Critical:

Weapon System Unique — *An operational task that is performed during the normal operation of the specific weapon system but not required by other weapon systems within its same general category (fighters, bombers, transports, SAM's, AAA, airborne ASW systems, etc.)*

Example: A step in a procedure to launch a missile from fighter 'A' is not required by any other fighter to launch that same type of missile.

Weapon System Mission Unique — *An operational task which because of the inherent demands of the missions (Air Interdiction, Close Air Support, Offensive Air, etc.), is performed as part of the normal operation of a specific weapon system.*

Example: Fighter 'A', in performance of a Close Air Support mission, must communicate with a Forward Air Controller via aircraft UHF radios in order to be cleared for attack on a target. This task is not required for any other type of mission assigned to this particular weapon system.

Weapon System Critical — *A task and/or a sequence of tasks for a specific weapon system that must be performed correctly in order that established operational safety standards are met.*

Example: Accomplishment of the appropriate immediate action procedure step for fighter 'A' when confronted with an engine fire condition.

Weapon System Mission Critical — *A task and/or a sequence of tasks for a specific weapon system that must be performed correctly in order that the objectives of the assigned missions are fulfilled.*

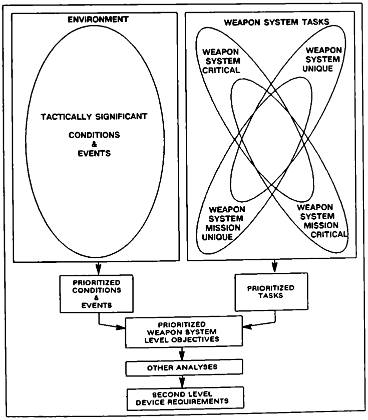

ENVIRONMENT
TACTICALLY SIGNIFICANT
CONDITIONS
&
EVENTS
WEAPON SYSTEM TASKS
WEAPON SYSTEM CRITICAL
WEAPON SYSTEM UNIQUE
WEAPON SYSTEM MISSION UNIQUE
WEAPON SYSTEM MISSION CRITICAL
PRIORITIZED CONDITIONS & EVENTS
PRIORITIZED TASKS
PRIORITIZED WEAPON SYSTEM LEVEL OBJECTIVES
OTHER ANALYSES
SECOND LEVEL DEVICE REQUIREMENTS

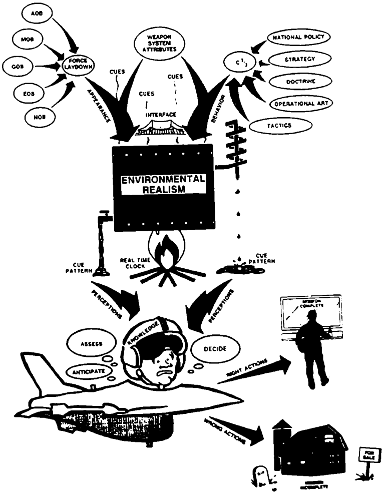
Fig. 8 Weapon System Description/Training Analysis Document

Example: Pilot of fighter 'A' must re-evaluate the planned target area formation tactics and modify and communicate as appropriate during the flight to meet the current tactical situation.

These categories are not mutually exclusive of one another. Some tasks may meet the definitions of all four categories, while others may satisfy three, two, or possibly only one of the definitions.

Herein lies a method of filtering and prioritizing tasks that aids the analyst in determining what is really important and where emphasis should be placed when defining device requirements. A task that qualifies in all four categories is deserving of special attention to ensure that the final design will satisfy the learning requirements. In like manner, tasks that qualify in fewer categories will generally carry correspondingly reduced levels of importance. This method is particularly useful when performing tradeoffs for cost versus learning value during system definition and design phases.

Depending on the intended purpose of the training device, it may not be necessary to filter the task list for all four categories. For example, if the purpose of the device is to train weapon system operation (procedural skills), the filters of greatest interest should be *Weapon System Unique* and *Weapon System Critical*. In like manner, if mission level training involving tactical decision skills is the focus for the device, then *Weapon System Mission Unique* and *Weapon System Mission Critical* tasks filters will be the filters of greatest importance. A device that is required to train all tasks pertinent to the operation and employment of the weapon system should be filtered for all four task categories. The use of analysts with extensive operational experience is necessary, in order that quality judgments are made as to the uniqueness and criticality of each task.



Summary

Central to this discussion has been understanding mission-ready crews' responsibilities and training needs. After listening to their demands and carefully analyzing their operational requirements, it was concluded that the environments the crew members have available to them presently do not adequately support their training needs. Further analysis revealed that, in general, developing decision skills, and specifically learning tactics, is what is required of MR crews. These requirements demand that more realism be present in the training environment. The level of realism required to support the instruction of tactics can only be achieved when both the appearance and behavioral aspects of the combat environment are properly represented. The simulations of OOB's and $C^{2/3}$ structures are the key environmental elements that will provide the appearance and behavioral qualities required for tactical realism. Particular care must be given to the modeling of $C^{2/3}$ such that the cues presented are valid. Special consideration must also be given to the current limitations of visual systems to produce the anticipated large quantity of dynamic cues. Measures of uniqueness and criticality are useful criteria to aid in the prioritization of tasks. Additionally, the measure of tactical significance must be imposed upon the entire realm of environmental cues so that cues that are truly important are displayed to the crew member.

Providing a realistic training environment is key to providing the capability to *train like they fight*. When considering peacetime constraints, a high fidelity electronically simulated environment holds the best promise for success. Technologies exist today that open the doors to new horizons for tactical training in simulation devices. The challenge to industry is to properly apply those technologies in such a way that a realistic combat environment is produced.

Acknowledgements

The author would like to acknowledge the thoughtful efforts of Winnie Miller, Mike Willmore, Dr. Ed Stark, and Pat Yearick for their constructive critiques and the proofreading of this paper. Additionally, I thank George Borst, for the outstanding art work.

References

1. Department of Defense Dictionary of Military and Associated Terms, Joint Chiefs of Staff Publication 1 (Washington, D.C., Government Printing Office, 1986).

2. AFM 1-1, Basic Aerospace Doctrine of the United States Air Force, (Headquarters, Department of the Air Force/XOXID).

3. Gen. Larry D. Welch, *NATO Rapid Reinforcement - A Key to Conventional Deterrence,* (NATO's Air Forces, Feb 1987).

4. D.C. Schultz, J.M. Owens, and S.D. Harris (1987), *Challenges to the Joint Services V-22 Osprey Total Training System.*

5. AF Form 723.

6. David B. Guralnik, ed., *Webster's New World Dictionary* (New York, Popular Library Inc.,1971).

7. Stanley N. Roscoe, "Visual Cue Requirements in Imaging Displays", in Proceedings of Image 77 Conference.

8. "Survive, Acquire, Destroy Tactics (SAD-TAC)", Strike Fighter Tactics Manual (Light Attack Weapons School, Pacific NAS, Lemoore, CA..

9. "TAC/AAC/PACAF/USAFEM 51-50 Vol I (Headquarters Tactical Air Command, Oct 1985).

APPENDIX A

VISUAL SCENE REQUIREMENTS

MOVING OBJECTS	AIR FORCE MISSIONS											CORRELATE WITH	
	STRATEGIC AEROSPACE		COUNTER AIR			AIR INTER-DICTION	CLOSE AIR SUPP.	SPEC OPER	AIR-LIFT	AEROSPACE			
	OFF	DEF	OFF	DEF	SEAD					SURV & RECON	MARIT. OPER.		
TANKS					●	●	●	●		●			RADAR IR SENSORS EW EMITTERS OTHER SENSORS
APC'S					●	●	●	●		●			
TRUCKS					●	●	●	●		●			
RADAR VANS					●	●	●	●		●			
SUPPORT EQUIPMENT					●	●	●	●		●			
SAM'S IN FLIGHT	●	●	●	●	●	●	●	●	●	●	●	FLIGHT PROFILE	
SAM'S ON TRANSPORTERS/LAUNCHERS					●	●	●	●		●			
AAA VEHICLES					●	●	●	●		●			
HELICOPTER (ENEMY AND FRIENDLY)	●	●	●	●	●	●	●	●	●	●	●		
AIRPLANES (ENEMY AND FRIENDLY)	●	●	●	●	●	●	●	●	●	●	●		
TRACERS IN FLIGHT	●	●	●	●	●	●	●	●	●	●	●	PROJECTILE TROJECTORY	
CHAFF	●	●	●	●	●	●	●	●	●	●	●		
FLARES	●	●	●	●	●	●	●	●	●	●	●		
BOMBS (HIGH DRAG - IN FLIGHT)			●	●			●	●					
RPV'S / DECOYS					●		●		●	●	●		
PARATROOPS							●		●	●			
ROCKETS							●			●		PROJECTILE TROJECTORY	
AIR TO AIR MISSLES	●	●	●	●	●	●	●	●	●	●	●	FLIGHT PROFILE	
AIR TO GROUND/SURFACE MISSLES	●	●	●	●	●	●	●	●	●	●	●		
TURRET ROTATION (TANK)							●						
MISSLE LAUNCHER ROTATION					●		●				●		
JETTISON OF AIRCRAFT STORES		●	●	●		●	●	●			●		
SURFACE SHIPS				●				●		●	●		
SUBMARINES										●	●		
SATELLITES	●	●	●	●				●		●	●		
MOBILE ICBMS	●	●								●			
INFLIGHT ICBMS	●	●								●		FLIGHT PROFILE	

VISUAL SPECIAL EFFECTS	AIR FORCE MISSIONS											CORRELATE WITH
	STRATEGIC AEROSPACE		COUNTER AIR			AIR INTER-DICTION	CLOSE AIR SUPP	SPEC OPER	AIR-LIFT	AEROSPACE		
	OFF	DEF	OFF	DEF	SEAD					SURV. & RECON	MARIT. OPER	
DYNAMIC SMOKE	●	●	●	●	●	●	●	●	●	●	●	WEAPON DETONATION TARGET HIT
DYNAMIC FIRE	●	●	●	●	●	●	●	●	●	●	●	
DRIFTING SMOKE	●	●	●	●	●	●	●	●	●	●	●	WINDS
LINGERING SMOKE	●	●	●	●	●	●	●	●	●	●	●	TIME
COLORED SMOKE	●	●	●	●	●	●	●	●	●	●	●	MOVING SMOKE SPECIFIC SAM/AAM/AGM
AIR BURST SMOKE	●	●	●	●	●	●	●	●	●	●	●	WEAPON DETONATION
AIRCRAFT ENGINE SMOKE TRAILS	●	●	●	●	●	●	●	●			●	INTESITY -ENGINE SPECIFIC -POWER SPECIFIC
GROUND VEHICLE ENGINE SMOKE					●	●	●	●				
MISSILE SMOKE TRAILS	●	●	●	●	●	●	●	●	●	●	●	MISSLE FLIGHT PATH
SMOKE RISE FROM OCCULTED SOURCE					●	●	●	●	●	●		SOURCE MOVEMENT WINDS
SMOKE RISE FROM POINT SOURCE	●		●	●	●	●	●	●	●	●	●	SAM/SSM LAUNCH SOURCE FIRE
GUN SMOKE (AIRCRAFT & ARTILLARY)	●		●	●	●	●	●	●		●	●	WEAPON FIRING WINDS/RELATIVE WIND
SMOKE GENERATORS ON TANKS AND VEHICLES					●	●	●	●				WINDS
DYNAMIC DUST	●				●	●	●	●	●			MOVING VEHICLES TERRAIN BULLET IMPACTS
DRIFTING DUST					●	●	●	●	●			WINDS
DUST RISE FROM OCCULTED SOURCE	●				●	●	●	●				SOURCE MOVEMENT WINDS
BLOWING SURFACE SAND					●	●	●	●	●			WINDS
BLOWING SURFACE SNOW					●	●	●	●	●			
HOVER WHITE-OUT								●				SNOW GROUND COVER AGL ALTITUDE
HOVER BROWN-OUT								●				
ROTOR DOWNWASH GROUND EFFECTS							●	●				AIRCRAFT SPECIFIC COLLECTIVE AIRCRAFT AGL ALTITUDE
ROTOR FLICKER							●	●				ROTOR SPEED
ENGINE CONTRAILS	●	●	●	●	●	●	●	●	●	●	●	ATMOSPHERIC CONDITIONS
WING FORM CONTRAILS		●	●	●	●	●	●	●			●	ATMOSPHERIC CONDITIONS
FUEL VENT CONTRAILS		●	●	●	●	●	●	●			●	AIRCRAFT G SPECIFIC AIRCRAFT
AFTERBURNER PUFFS		●	●	●	●	●	●	●			●	SPECIFIC AIRCRAFT AFTERBURNER SELECTION
SCATTERED DRIFTING CLOUD DECK	●	●	●	●	●	●	●	●	●	●	●	WINDS
BROKEN DRIFTING CLOUD DECK	●	●	●	●	●	●	●	●	●	●	●	
LAYERED CLOUD DECKS	●	●	●	●	●	●	●	●	●	●	●	
DISTANT CLOUD BUILD-UPS (TSTMS)	●	●	●	●	●	●	●	●	●	●	●	RADAR .SENORS
CLOUD SHADOWS ON SURFACE	●	●	●	●	●	●	●	●	●	●	●	SUN POSITION WINDS
RAIN ON WINDSCREEN-ENTIRE FOV (OUTSIDE OF CLOUDS)	●	●	●	●	●	●	●	●	●	●	●	AURAL CUES
RAIN SHOWERS / SHAFTS IN DISTANCE	●	●	●	●	●	●	●	●	●	●	●	RADAR .SENORS
WINDSCREEN RAIN REMOVAL	●	●	●	●	●	●	●	●	●	●	●	COCKPIT SWITCH POSITION LOCALIZED EFFECT ON WINDSCREEN
OWNSHIP VISIBLE IN FOV	●	●	●	●	●	●	●	●	●	●	●	CREW MEMBER - SIGHT DIRECTION
STRUCTURAL ICE	●	●	●	●	●	●	●	●	●	●	●	CREW MEMBER - SIGHT DIRECTION ATMOSPHERIC CONDITIONS AIRCRAFT SPECIFIC
OWNERSHIP WING WHITE-OUT		●	●	●	●	●	●	●			●	AIRCRAFT SPECIFIC AIRCRAFT G ATMOSPHERIC CONDITIONS
FLIGHT CONTROL DEFLECTIONS		●	●	●	●	●	●	●			●	OWNSHIP OTHERSHIP
MISSILE SEPARATIONS FROM A/C	●	●	●	●	●	●	●	●			●	
WEAPON DETONATIONS	●	●	●	●	●	●	●	●	●	●	●	WEAPON TYPE -MISSILES -BOMBS -CBU'S -BULLETS
SECONDARY EXPLOSIONS	●				●	●	●	●			●	TYPE TARGET
WEAPON DAMAGE EFFECTS	●	●	●	●	●	●	●	●			●	TYPE TARGET TYPE WEAPON TYPE DELIVERY ACCURACY OF DELIVERY
MUZZLE BLAST (DAY/NIGHT)					●	●	●	●			●	TIME OF DAY TYPE OF WEAPON
BULLET RIPPLE (LOCALIZED VISUAL DISTORTIONS)							●					BULLET FLIGHT PATH
SUN GLINTS		●	●	●	●	●	●	●			●	SUN POSITION OBJECT ORIENTATION
DAY AFTERBURNER GLOW		●	●	●	●	●	●	●			●	ENGINE SPECIFIC OWNSHIP IR SENSORS
NIGHT AFTERBURNER PLUME		●	●	●	●	●	●	●			●	
OWNERSHIP ANTICOLLISION STROBE LIGHT EFFECT	●	●	●	●	●	●	●	●	●	●	●	TIME OF DAY VISIBILITY
'IN' SUN AND 'DOWN' SUN VISIBILITY EFFECT		●	●	●	●	●	●	●			●	SUN POSITION CREW MEMBER SIGHT DIRECTION
MOON		●	●	●	●	●	●	●	●		●	
VISUAL IDENTIFICATION LIGHTS		●										

89-3302-CP

FIDELITY REQUIREMENTS IN AVIATOR TRAINING NETWORKS

Gary R. George*, Samuel N. Knight and Edward A. Stark*

CAE-Link Corporation
Link Flight Simulation Division
Binghamton, New York

ABSTRACT

Networks of limited-fidelity training devices have demonstrated their value in developing proficiency in many interactive and tactical skills for a number of system crews, and for personnel having command and control responsibilities, but the dynamics of the unit tasks and interactions are being supported by levels of fidelity and information update rates which are inconsistent with the tactical responsibilities of aviation units and aircrews. Here, accurate, rapid decisions require high levels of tactical information as well as rapid and accurate updating of that information. Recent experience in the networking of full-fidelity flight simulators has shown that systems can be created to support training in advanced, dynamic interactive and decision-making skills for aviation crews using new capabilities for the realistic modeling of threat characteristics, doctrine and performance on a global level. Networks of high-fidelity simulators need to be expanded in the preparation of combat-ready aviation crews. Provisions are also needed to interface these high-fidelity networks with lower level devices and networks to support training in total combined arms operations.

INTRODUCTION

Individuals and crews operating on future battlefields will have to be proficient in a wide assortment of tasks and skills. Most involve high levels of tactical decision-making under heavy crew workloads, induced in part by a multiplicity of both friendly and hostile battlefield elements. Most individual and crew tasks are well supported in a variety of simulators and special function trainers, but the skills required in team, element, and unit interactions, involving tactical coordination and tactical decision-making are not yet receiving adequate support, especially in combat aviation.

Warfighting makes heavy demands on the skills of individuals and crews, but warfighting is a team process, requiring the coordinated performance of individuals and crews having unique, but mutually-supporting capabilities and responsibilities.

Courtice, in a recent analysis[1] of combat training requirements has described combat as highly dynamic, and highly variable, requiring the deliberate and systematic training of individual skills, but especially important, training in the real-time integration of those skills.

Hughes[9] in a paper on embedded training has noted that the tactical system itself, operating in the real-world, can support some of the skill integration function, but not all of it. High-fidelity simulators, and networks of simulators as well as embedded training systems are needed, because they can supply critical tactical information which cannot be supplied otherwise, even in real-world practice.

Today increased use is being made of networks of simulators and training devices with integrated battle control stations and interactive threat simulations, in support of integrated team training. These networks are contributing to the expansion of combat skills training beyond traditional individual and crew practice and one-against-the-world scenarios, permitting combat teams to "train like they will fight," in complex, dynamic and inherently flexible battlefield exercises.

Thorpe[12 13] has reported on the use of SIMNET, a network of simulators used in training ground forces in combined arms operations. George, et al[6 7] have described the training of helicopter crews employing networks of flight simulators in combat team operations. Each of these efforts has required the analysis of new sets of team training objectives whose implications for modern training and instructional technology are not yet well understood.

SIMNET[13] has proven to be highly effective in meeting many of the training objectives associated with ground combat, using minimum or selective levels of simulator fidelity. MULTISIM[6 7] has been found effective in supporting the tactical training of helicopter teams, but employing much higher levels of simulator fidelity to support the more stringent training objectives associated with aviator team operations. Training economy, as well as training efficiency and training effectiveness, require that training be accomplished in the lowest-cost systems possible, but this requires a thorough analysis of the training objectives, and of the learning processes to be supported in defining the system's fidelity requirements.

Much has been written regarding the fidelity levels needed for effective simulator training. Hays and Singer[8] have published an excellent summary of the research on the determination of fidelity requirements in the design of training systems. They have also provided a set of guidelines for determining the degree of fidelity needed in a given training situation, or training device. The training analyst begins by evaluating the tasks to be trained, and identifies the information required, in the training device, for effective training. The type and complexity of the information (and fidelity level) required depends on the nature and complexity of the information available to, and needed by the operator in the real task setting. For team training, and in the training of related elements and units, the personnel themselves provide much of the infomation required in support of performance and learning. In addition, aviation team operations, especially, take place in complex, rapidly changing environments where complex discriminations, interpretations, and judgments must be made as the mission progresses, adding further to the information needed in task practice.

A paper presented at a symposium on simulator fidelity requirements is of special interest in establishing the development of the setting for team training. Eggemeier and

* Member AIAA

Cream5 found that it was necessary to expand the analysis of individual tasks performed by members of a weapon system team to account specifically for team interactions. In effect each member of the team was found to be a critical source of information for each other member. Montero and his colleagues11 documented the major changes taking place during the training of weapon system teams. These changes reflect, primarily, the development of skills in anticipating, collecting, supplying and responding to information generated by members of the team.

The integral relationship between training objectives and system fidelity requirements is illustrated in Figure 1. The required fidelity level, in essence, is a reflection of the complexity of the training setting, including the system within which the crew trains and the information influencing crew responses. Repeated experience in obtaining successful responses instills, builds up and fortifies the crews' tactical skills which, in effect is the goal of the training objectives for which the training setting was established.

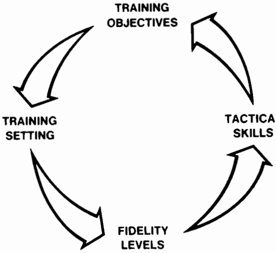


Figure 1 Training Objectives and Fidelity

A skill is the ability to perform a task or series of tasks under a particular set of circumstances. It is characterized by the collection and processing of task information, and results, eventually, in observable performance. A key element of the training setting which strongly affects the fidelity requirements is the information content which must be available to properly support the needed information processing to fulfill a specific training objective.

For example, a pilot learnıng to hover a helicopter needs to have information about the location of the horizon, his heading, altitude and altitude rate. nosition of the helicopter with respect to a position on the ground, and information about the positions of, and the helicopter's response to the controls. The skill, for the layman, is not simple, but experience indicates that it can be develop.d in a relatively simple low-fidelity training setting.

The same pilot learning to hover in a restricted area, on a slope, in the middle of an active battle area requires much more information, both to perform effectively, and to learn. Learning this second skill involves more information and more complex information processing on the part of the pilot and, thus requires a more complex training setting, or system of training settings than does the first example. Needless to say, team tactical operations involve even more complex information processing requirements.

Dees2 has reviewed the problem of training helicopter crews in air-to-air combat, and the simulation technology available to support such training. He has concluded that networks consisting of at least two simulators, each capable of operating according to U.S. or threat doctrine are essential. He has also noted that, "The more competent the aviator, the more sophisticated must be the simulator." In effect, the higher the skill level of the crew member being trained, the more information they can process as they refine and extend those skills in team and combined arms settings.

Information Processing in Team Tactics

While skilled human performance is often characterized in terms of overt observable motor activity, it is comprised primarily of the covert, unobservable processing of information which results sometimes, but not always, in observable behavior.

Both motor and information-processing skills are developed through learning; learning is supported by training settings which provide the information the trainee must learn to sense, perceive, anticipate, interpret, and employ in performing the required missions, tasks, and subtasks.

Krey, in a recent paper10 on crew coordination training, emphasized the need to teach crew members to recognize changes in the situation, and to understand the potential impact of those changes on the mission, and on the personnel and systems involved. Tactical situations are characterized by change, and much of the change is clearly identifiable, but much of it is subtle, requiring skill in recognizing and processing information in many different forms, having many degrees of significance.

The information needed to facilitate tactical team task performance and skill learning originates in at least seven kinds of sources: **The system** being employed in performing assigned mission functions provides much of the information needed in tactical performance. It may take the form of visual displays, system sounds, crew station motion, or the unique resistance of controls to their operation. The **visual environment** external to the system, observed through unaided vision of terrain, vegetative, and cultural features and activity outside the crew station provides much of the more important tactical information. Visual displays of information generated by specialized **sensor systems** including radar, infrared, and television or optical systems are redundant with some information observed in direct vision, but also provide a significant amount of unique stand-off tactical information. The fourth source of information is in the radio and **communications systems** which provide information from outside the crew station. **Other member(s) of the crew** in multiple crew stations provide some of the most important tactical information as do the personnel making up **the team** with which the crew interacts. Finally, the **crew member himself** supplies information from his own memory in the form of doctrine, principles, performance standards, procedures, expectations and from his own movements and actions

The information impinging on crews from these various sources is processed in a number of ways, depending on the task being supported. The development of skills in information processing can required extensive and varied practice. Traditionally, training settings tend to be designed to support one or few of these processes at a time. For tactical teams, proficient in the operation of their vehicles of war and their various systems, training in the simultaneous processing of four kinds of information is especially important.

1. Team Procedures

The operations required of the members of a tactical team are usually segmented in sequences of activities involving corresponding sets of tactical functions. Standard procedures are developed to form the framework for organizing each function in the sequence of required operations. Each procedure tells each individual in the team what he must do and when he must do it, but, equally important, it tells each team member what each other team member will, or should do. Many tactical procedures are memorized, but they also need to be practiced in a team environment to ensure the coordination required in tactical operations. Procedures are memorized and applied as more or less fixed sequences of actions, but they are predicated on two other important processing capabilities - recognition and anticipation.

2. Recognition

Each and every team member must recognize and discriminate among an almost infinite variety of patterns of information in deciding what they require of him, and in anticipating their potential impact on him, on the team, and on the mission. The simplest recognition skills relate to the well-defined events and conditions under which specific procedures must be applied, modified or terminated. The most complex involve discriminating among patterns of information about the terrain, the weather, threats, friendly elements, and the deployment and behavior of both friendly and hostile forces. Each of these levels of recognition and discrimination requires practice, first in making accurate assessments of their meaning and, second, in discriminating among dynamic events in real time while performing other procedures, maneuvers and tactical processes under time pressure and stress.

Recognition skills are especially crucial to the aviator's tactical proficiency, and they are especially difficult because of the number of information sources typically available, the amount of information to be processed, limits in the time available, and perhaps most important, limits in the quality and the completeness of the information usually available in combat operations. Recognition and discrimination must almost always be accomplished with fleeting,fragmentary and frequently conflicting information. Much of the crew's information arrives by way of radio and sensor systems which may be degraded by weather, jamming, and other incidental and intentional effects. Information received through direct vision is also subject to limits imposed by weather, camouflage, the use of cover and concealment on both sides of the conflict, and by the need to perform recognition tasks at maximum range with minimum cues.

In team operations much of the information needing to be recognized and discriminated is in the performance of other members of the team and of the other tactical elements being supported by, or providing support to the operation. Skilled performance requires that each team member know and do the right things at the right time, as well as knowing what to expect of others - what they should do and what they probably will do, and what they really do, in actual practice.

3. Anticipation

The most important task of a team is to "stay ahead " of the situation. Detailed mission planning anticipates all likely contingencies and prepares the crew to respond effectively to them, but some events and circumstances having potential impact on the crew and its mission are unanticipated, requiring rapid reorientation and response. Skill in anticipating tactical events is hard to develop because of the difficulty in organizing practice in complex, widely varying tactical situations, but it is essential in even minimal levels of combat proficiency.

Success in tactical operations requires skill in anticipating the behavior of other crew and team members, and of other friendly and hostile elements on, over, and around the battlefield.

4. Interpretation

The information available to a crew in planning sessions, briefings, in their own experience, and in the events and activities occurring during mission operations provide the basis for the selection of a course of action to be employed in accomplishing the assigned mission. Before a course of action can be initiated the mass of available information must be evaluated, prioritized, interpolated, and extrapolated in accordance with tactical doctrine and experience. Time permitting, a number of possible alternatives are developed, evaluated, weighed and, if possible, rehearsed before one is selected for implementation. The process of interpretation is also very difficult because it involves as much information as is available as well as confidence, as acquired through experience, in anticipating the chains of events and their consequences, resulting from the implementation of each possible tactical alternative.

The Aviator's Added Dimension

The aviator like other members of the combined arms team needs to process a great deal of information, but the aviator's operating environment has an added dimension, height above the terrain. This and his mobility on the battlefield, force him to learn to perform with three distinct differences over other ground based elements:

1) The amount of information the aviator has to process is larger due to aircraft complexity, and his larger picture of a complex battlefield environment.

2) The processing of that information must be accomplished much more quickly due to the high mobility of the aircraft and its threats.

3) The accuracy of the aviator's tactical decision making must be correct the first time, even under the time pressures typical of air combat.

This means high workloads under stressful conditions, near information overload with time as a critical variable. An error in decision making or situational awareness can result quickly in catastrophe. This is not meant to imply that a decision-making error in armor, for example cannot have a tragic end. The important point is that the aviator tends to have less time to correct an error as well as having another dimension, and another time frame within which to make it.

The complexity of today's advanced aircraft also contributes to the fidelity requirements for aviator team training. Today's aviator must not only be an excellent pilot able to fly nap-of-the earth, and effectively conduct air-to-air combat but he must also be an avionics and threat expert, accomplished in communication and navigation skills, be a master of his weapon systems and be fully knowledgeable in doctrine as it applies to him and to his team. The fact that the aviator or aviation crew, through progressive training, becomes proficient in the operation of a particular aircraft before beginning team training, does not preclude the need for full-fidelity aircraft and weapon systems simulations at the team training level. On the contrary, the concept of progressive training requires that team training enhance the aviator's previously acquired skills by adding to the workload previously mastered. Since the aircraft and its systems are primary contributors to the crew's workload and performance potential, changing the workload for team training could negatively affect the resulting learning process.

The level of simulator fidelity is the simulation sophistication necessary to supply information required in practicing tasks performed in a specific mission. For the aviator this means that the simulator must be concurrent with his aircraft; that it flies and interacts with the environment as expected in the real-world, and perhaps most importantly, that the simulator encourages the aviator to think and react as through he were in a combat environment.

High Fidelity Networks

It is conceivable that the entire range of skills required in even the most complex tasks could be developed in a single training setting, but training efficiency, cost, and even learning efficiency require that most skills be developed in systems of progressive training settings, each geared to a definable set of tasks, skills, or skill elements, partly to avoid the expense of storing, controlling, and displaying large quantities of information, but also to avoid overloading the trainee's information-processing capacity in any given step of the skill-learning process.

Much of the information, the task loading, and the information update rates needed to support the integration of aviation crew and team skills, and the development of tactical team proficiency could be provided in real-world flight operations. However, in most instances, this is prohibitively expensive and hazardous and except for very limited exercises, it is also extremely difficult to schedule. As Hughes[9] has pointed out, some team skills, and especially those relating to electronic warfare, could be integrated and perfected in actual flight operations, using embedded training systems. Hughes has also noted, however, that recent advances in computer image generation in particular make the simulator the logical place to integrate team and unit skills under combat-like conditions.

The combat missions assigned to combined arms units require very high levels of proficiency in a large number of individual, crew, and team skills - in some units, more kinds of skills than in others. While those individual and crew skills can be developed in a variety of traditional training devices, and many team skills can be developed in special-function settings and others in real-world practice, ultimately, high-fidelity (high information content) networks are needed for some members of the overall team.

Simulators such as the AH64 Combat Mission Simulator provide training for Army aviation crews with high task loading resulting from realistic avionics and system simulations[3] [4] interacting with a realistic threat. Networks of simulators like the AH-64 Combat Mission Simulator could be used to teach helicopter teams to survive in an environment populated by realistic, intelligent, and interactive threats.

An example of a specific requirement for high fidelity simulator networking is training in the advanced phases of air-to-air combat which requires a complex array of information processing and tactical systems skills. In scenarios that are often measured in seconds, the crew must be proficient in the intricacies of engaging one or more equally determined threats while, at the same time, remaining totally aware of numerous other equally lethal elements including the terrain, cultural obstacles, ground threats and other teammates. Networking of high fidelity simulators is an excellent way to train these difficult tasks; it allows flying the aircraft to its limits to survive and to gain an advantage on the opponent. It is important that the information reflecting the "limits," and the overall flight and weapons characteristics of the aircraft be exactly like those experienced in the aircraft itself, as is the case in full-fidelity devices. Lower fidelity limits the ability of the pilot to know or to be able to use the aircraft's full tactical capabilities to his advantage and in some instances might allow the crew to learn techniques that could not be technically or safely employed in the actual aircraft.

The coupling of full-fidelity training devices requires a high-fidelity network architecture which can interface large volumes of data and high speed data rates to fully support the interactions of the individual devices at a level consistent with the fidelity of those devices.

A high speed network is needed to support the high computer update rates required in the presentation of visual cues in the various simulators. Air-to-air combat involves extremely high rates of aircraft maneuvering. Low visual system update rates result in stepping, causing eyestrain, performance degradation, general fatigue, and in some cases, simulator sickness. The interface between the simulators must be of sufficient speed to allow the state of each simulator and the various special visual effects to be updated at a high rate so that the visual display in participating simulators does not jump. Methods using extrapolation or high-pass filters can be used to smooth this effect for small step sizes. However, large step sizes (i.e., low update rates) cause uncertainty regarding aircraft state and position, particularly at high maneuver rates. The resulting errors adversely affect the crews response especially in low-level formation flight and air-to-air combat. The errors can also affect the simulated performance of advanced weapons such as the Hellfire missile.

Combined Networks for Synthesized Battle Exercises

Each member of the combined arms team has requirements for exposure to a hierarchy of training resources. Training starts in the classroom and works upward on a pyramid comprised of increasingly sophisticated training devices, each designed to support a specific set of training objectives within that hierarchy. Figure 2 shows the aviation pyramid with a network of more advanced training devices at the top. Aviation requires high fidelity networks, again, because of the magnitude, the complexity, and the variety of information to be processed, and the rates at which that information changes.

To optimize training in combined arms interactions however, interfaces must be employed to link the full-fidelity network with lower fidelity or selective fidelity devices such as

SIMNET. These interfaces should be designed at fidelity levels consistent with the skills exercised in SIMNET, and with the aviation skills to be integrated. Advanced networks should also contain a force level simulation which provides doctrine and the exact tactics of threats of various sizes and combinations, including global, 3-sided, or regional forces. Figure 3 shows the interconnection of the various training pyramids with the force level simulation. A potential addition to the network is a resource that few have considered, embedded training devices. Since future military procurements of equipment such as the LHX will require embedded training this is a logical addition to a network to utilize all training options.

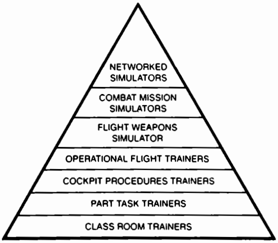


Figure 2 The Aviation Training Pyramid

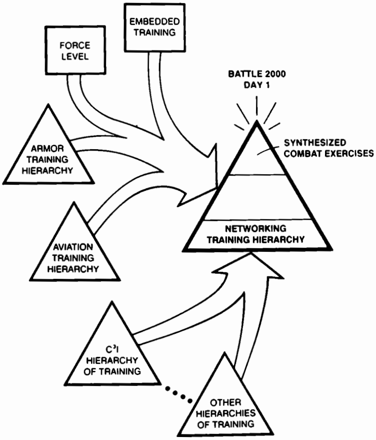

FORCE LEVEL
EMBEDDED TRAINING
BATTLE 2000 DAY 1
SYNTHESIZED COMBAT EXERCISES
ARMOR TRAINING HIERARCHY
AVIATION TRAINING HIERARCHY
NETWORKING TRAINING HIERARCHY
C³I HIERARCHY OF TRAINING
OTHER HIERARCHIES OF TRAINING

Figure 3 Network Training Pyramid

Full fidelity networks with their higher rates and advanced simulations, necessary for aviation tactical training, can interface with the lower or selective fidelity networks and embedded training devices to provide effective combined arms and joint force training. These networks can provide essential player interactions at levels of fidelity which are optimized for the systems, the tasks and the missions begin supported. The incorporation of force level simulation has a major logistical advantage, since setting up networks of hundreds of simulators on various long and short haul networks will take significant coordination and effort, comparable to that experienced in a real world battle. This cannot be tolerated where time is of the essence in training. Force level simulation allows smaller networks of teams to participate in scenarios involving thousands of realistic players.

The ideal network of full-fidelity and selective fidelity simulators, combined with force level and embedded training provides the synthesized battlefield needed for tactical skill integration, minimizing the need for training in actual field exercises, drastically increasing the effectiveness of tactical training, while minimizing training costs. Every attempt should be made to use existing devices whenever possible, as in the case of MULTISIM6 7 since networking technology is relatively low cost compared to the procurement of new devices.

In Figure 3, Synthesized Combat Exercises represents the top of the networked training system hierarchy. The top of the pyramid is as close as training can approach to the first day of an actual battle. In the first few days of battle, the learning experienced by the survivors historically greatly increased their chances of surviving, and winning, in subsequent engagements, but the battlefield is the wrong place to learn to fight. No one dies or is injured in the synthesized battle environment described here. Also, evaluation in the form of automated scoring, records of overall mission performance, and record/playback can be used for research as well as for training.

CONCLUSION

Aviation crews require high fidelity simulations not only in individual flight and weapon system trainers, but also in the networks used in generating the information necessary for the crew to develop proficiency in team operations.

Synthetic combat environments should also be created by combining low, selective, and full-fidelity networks with force level simulators and embedded training systems. This would allow each specific discipline of the overall team to operate at the fidelity levels appropriate to its respective real-world mission tasks and workloads. This would also provide greatly expanded capabilities for team training for combined arms elements as well as for joint forces.

Anything less will result in training objectives being unfulfilled and a considerably reduced state of mission readiness.

REFERENCES

1. Courtice, A. J.; COMBAT TRAINING - THE NEXT FRONTIER OF AIR FORCE TRAINING TECHNOLOGY; Interservice/Industry Training Systems Conference, 1988.

2. Dees, J. W.: ESSENTIAL CONSIDERATIONS FOR SIMULATION OF AIR TO AIR COMBAT; United States Army Aviation Center, Ft. Rucker, Alabama, 1 November 1985.

3. Drew, E.W., George, G.R., Knight, S.N., AH-64 COMBAT MISSION SIMULATOR TACTICAL SYSTEM, proceedings of the AIAA Flight Simulation Conference, Monterey, CA, August 1987.

4. Drew, E.W., George, G.R., Knight, S.N., VISIONICS SIMULATION IN THE AH-64 COMBAT MISSION SIMULATOR, proceedings of the National Aerospace and Electronics Conference, Dayton, OH May 1988.

5. Eggemeier, F.T. and Cream, B.W., SOME CONSIDERATIONS IN THE DEVELOPMENT OF TEAM TRAINING DEVICES, in Erwin, D.E. (Ed.), Psychological Fidelity in Simulated Work Environments, Proceedings of a Symposium, American Psychological Association, Toronto, August 1978.

6. George, G. R., Knight, S.N., and Monette, R.; MULTIPLE SIMULATOR NETWORKING (MULTISIM) - THE WAY TO PROVIDE EFFECTIVE COMBAT TRAINING TODAY; Interservice/Industry Training Systems Conference, Orlando, FL, 1988.

7. George, G. R., Knight, S.N., and McTiernan, J.; NETWORKING OF FULL FIDELITY SIMULATORS FOR ADVANCED MISSION TRAINING; Proceedings of the National Aerospace and Electronics Conference, Dayton, OH, May 1989.

8. Hays, R. T. and Singer, M. J.; SIMULATION FIDELITY IN TRAINING SYSTEM DESIGN; Springer - Verlag, New York 1988.

9. Hughes, R. G., AIRCREW EMBEDDED TRAINING; IEEE AES Magazine, September 1988.

10. Krey, N. C.; USING AIRCREW COORDINATION TRAINING TECHNIQUES TO ENHANCE WAR FIGHTING CAPABILITIES; Interservice/Industry Training Systems Conference, Orlando, FL 1988.

11. Montero, R. C., Campbell, W. J., Zimmer, S. and Glickman, A.S., CHANGES IN TEAM BEHAVING DURING OPERATIONAL TRAINING, in, Proceedings of the Human Factors Society Annual Meeting, New York, October 1987.

12. Thorpe, J. A.; THE NEW TECHNOLOGY OF LARGE-SCALE SIMULATOR NETWORKING: IMPLICATIONS FOR MASTERING THE ART OF WAR FIGHTING CAPABILITIES; Interservice/Industry Training Systems Conference, Washington, DC, 1987.

13. Thorpe, J. A.; WAR FIGHTING WITH SIMNET - A REPORT FROM THE FRONT; Interservice/Industry Training Systems Conference, Orlando, FL, 1988.

FLEET REQUIREMENTS FOR F-14D AIRCREW TRAINER SUITE

Commander Bruce H. Hart, USN
Commander, Fighter Airborne Early Warning Wing, Pacific
Naval Air Station Miramar, CA 92145-5600

ABSTRACT

The U.S. Navy fleet fighter community needs simulators to conduct training which would otherwise be impossible due to peacetime safety constraints, the high costs of duplicating multiple enemy platforms, and the risk of compromising highly secret and sensitive systems. Additionally, simulators improve our capability to analyze mission training and performance. To accomplish this, however, certain areas of the simulation, simulation control, and simulation analysis tools must be improved beyond what is currently available. The F-14D Aircrew Trainer Suite is comprised of five, domed simulators which can be linked into one battle problem. Missions can be recorded and debriefed in a separate, collocated facility. The environment in which the battle problems are conducted is controlled by a separate computer and control system designated the Tactical Environment System. From a library of over 400 threats, 160 can be inserted into any one battle problem. The aircrew will be presented an environment with visual, television, infrared, and own/enemy radar signals correlated to constitute a "real world". In this system, all sensors are simulated, not stimulated. The fleet is on the threshold of being able to conduct and analyze training which would be otherwise impossible. There are areas of threat performance and environmental fidelity for which modeling must be developed to give the aircrews a realistic workload in order to train not only tasks, but also task management. Development of adequate controlling tools is essential to successful training. And, we must determine how much fidelity is required to meet the training requirement.

Background

Why Complex Aircrew Training Suites (ATS). In this era of reduced defense spending and limited training opportunities, the multidomed, high fidelity simulator offers a means to develop advanced tactics training against the projected threat. Until recently, in the fighter community, there was no way other than major fleet exercises for fighter aircrew to train against large scale attacks. The F-14D ATS promises exactly this capability. It is designed to give the Navy fighter community a realistic day and night time environment for up to five F-14D's. It is this potential which must be furthered. Up to this point in the computer industry, there was no ability to go beyond developing simulators which only train the crew in the basics. In the F-14D ATS and other advanced simulators, we can go beyond teaching students fundamentals. We can take an experienced crew and train them for fleet defense. This requires us to monitor computer advances, develop simulator potential, and meet evolving training requirements.

Uniqueness of Simulators vs Real World. This is not a call to exchange flight time for simulator time. In fact if the ATS is good and widely used, a fleet crew will only get an average of 1-2 hours per month in the ATS. But, there are increasing shortcomings in real world, multi-airplane exercises. In the real world:

1. We cannot get the quantity or quality of enemy equipment we will face in real combat.
2. We cannot extract data to answer the hard questions as to why something did or did not happen.
3. We cannot replicate any scenario.
4. We cannot precisely rerun a flight to establish probabilities, or see sensitivities to changed variables.
5. We cannot violate safety: We have large safety margins to avoid clouds, the ground and other aircraft, especially at night. They are the rules for training, but not for combat.
6. There is an increasing concern about security. While training fighter crews for the 21st century, we'll be increasingly concerned about keeping information from adversaries. It's too easy in our country for land or sea listening platforms to intercept the signals which describe our operations and tactics. Already there are rumors of programs we do not want to field because of the fear that our slight, technical edge will be lost. Can you imagine a boxer either not training, or training in an unlit room?

With the proper development of the modern ATS, all these can be overcome. The fleet is greeting the potential of these suites with tremendous enthusiasm.

The F-14D ATS consists of five cockpits which can be linked into one battle problem. Two of the cockpits, called the Weapon System Trainer (WST), are each enclosed in a 40 foot diameter dome which provides the pilot and Radar Intercept Officer (RIO) (hereafter referred to together as "a crew") with a presentation of their total, visual environment. The remaining three cockpits, Mission Flight Trainers (MFT), are each enclosed in a 30 foot diameter dome. An MFT's visual presentation is limited to a fixed area, 150 degrees horizontal by 45 degrees vertical. There are no motion bases associated with these five simulators, but motion cues are provided by the visual environment and a g-cueing system built into the ejection seat.

The remaining areas of the ATS used by instructors or aircrew are the Instructor Operating Stations (IOS) collocated with each dome, the Tactical Environment System (TES), and the Debrief Station. The TES is a separate control station which is required when any dome is linked with any MFT dome. The TES will contain a library of over 160 platforms. This is a formidable array of threats, which is quite adequate for training. These platforms can be either computer or human controlled. Each airborne platform is to be appropriately modeled for aerodynamic performance, radar and infrared signatures, and weapon system capabilities. Sensors will be simulated, not stimulated. The tactics for the computer driven platforms (CDP) will be developed from threat source documents. Variations of performance will be afforded by assignment four skill levels and limited use of Monte Carlo techniques. The Debrief Station has multiple displays and is dedicated to this function in the ATS to insure that trainer utilization is not adversely impacted by extended debriefs, and that displays and seating for debrief of multi-dome missions are available.

The Commander Fighter Airborne Early Warning Wing, Pacific, F-14D Fleet Introduction Team (FIT) hosted a meeting 6-7 April 1988, to determine the requirements for the ATS when up to five domes are linked together in a training or tactical development scenario. The FIT requested attendance of personnel from disciplines outside the fleet fighter community. Supporting this meeting were representatives from Tactical Training Group Pacific, Point Loma; Center for Naval Analysis; Weapons Tactics Analysis Center, China Lake; the chain of command, and F-14 and E-2 Fleet Replacement Squadrons (FRS) from NAS Miramar.

The approach taken was to examine the operational requirement for the ATS objectives somewhat constrained by the hardware of the simulators, the associated IOS and the TES. Although representatives of Grumman Electronic Systems (prime contractor for the ATS) and Naval Training Systems Center (NTSC) attended, the requirements meeting was not constrained by consideration of what might or might not be in scope to the current ATS/TES contract.

This paper summarizes the results of that meeting. It presents what ATS features are required based on the ATS mission. The categories into which the requirements have been broken are: Fidelity, Scenario Generation, Control and Data Extraction.

<u>Fidelity</u>

Is there enough fidelity in our simulator system that what we learn and practice in the simulator will be transferable to the real world of combat? Will our training be valid?

<u>Visual Fidelity.</u> Why be concerned with the visual environment or visual fidelity in these areas? Possibly some might not conceive that fighter pilots will have visual engagements or dogfights in the future. The Advanced Tactical Fighter (ATF) concept, however, is that of low observables--that is hiding from enemy radar and infrared sensors. This implies that tactical aircraft weapons employment ranges will shrink back to visual ranges once more, and that the gun and the visual dogfight are still highly probable conclusions to an engagement between stealth, fighter aircraft.

<u>Threat Aircraft.</u> An indicator of our concern is that 32 different aircraft have been selected to be modeled. Various intelligence agencies have been contacted to either directly or indirectly supply intelligence for these models. This has included the Foreign Technology Division at Wright Patterson AFB, which has the charter to provide all data that models any aspect or feature of airborne platforms. Airborne platforms include aircraft, air-to-air and air-to-ground missiles. The data includes the total spectrum of aerodynamic performance, electronic warfare, both passive and active; radar cross section; and infrared signature for various frequency bands of infrared energy. The fleet concern is that what is presented in the domes may vary significantly from the real world. And, worse yet, our crews may not have any idea of the types or the sizes of the variances.

<u>Modeling.</u> Intelligence data is collected at a sensitive or top secret level which cannot be disseminated to the operational fleet directly. However, purposeful desensitizing of data for a secret level simulator is not required. There are four areas in the modeling process in which fidelity is generally lost.

1. Data can be analytically developed from mensuration of engine inlets, wing shapes, human intelligence, etc, but the priority has not been there to force its calculation and formatting into a form suitable for simulators. As a result, estimates, or pure geometrical solutions are provided in the simulation. Providing a maximum sustained turn rate as the maximum possible turn rate would be an example of this problem.

2. A spectrum of mathematical modeling, ranging from minimal processor capability and memory requirements to sophisticated, real time solids modeling requiring the speed and processing capability of the Cray. Though greater processing power is desired for many models, economic constraints must be considered. The computer power, architecture, and memory to run models of adequately high fidelity need to be carefully determined and evaluated. Then, appropriate hardware must be specified and purchased.

3. You can lose fidelity from the real world in the resolution of the data table the math model uses. You can end up smoothing out critical discontinuities or subtleties in performance which a fighter pilot must learn to exploit to win engagements. An example of this was the small differential in performance in certain areas between the U.S. F-4 and the Soviet MIG-21 when they fought in the skies over Vietnam. The subtleties had to be exploited for victory.

4. If we do not have reliable data, pure estimates are made based on similarities to US technology.

As a result, how can the fleet know the level of fidelity of the simulated enemy?

Threat Aircraft Aerodynamics can be used as an example of the areas for which the difference from real world must be quantified and made known to trainees.

A U.S. fighter community was given a simulator model of an "enemy fighter" against which they developed specific disengagement or "bug out" tactics. These tactics were promulgated to their operational units. Later, a more sophisticated Air Combat Maneuvering (ACM) simulator was used to test this tactic against a higher fidelity model of the same enemy aircraft. The tactic was a disaster. The enemy, when modeled at higher fidelity, had better performance than the U.S. fighter in this aerodynamic/weapon envelope area. As the story goes, it turned out that the lower fidelity model of the enemy aircraft that had been used was a "tweaked" model of the friendly fighter.

5. In many cases, unless specifically requested, the threat fighter pitch rate and acceleration are not modeled from an intelligence estimate, but merely follow the pitch rates and accelerations which would be expected from the general, aerodynamic performance of that kind of vehicle. Therefore, if the aircraft can generate an instantaneous angle of attack to point its nose and shoot a weapon, but this is not sustainable, it is not shown in the turn rate tables. As a consequence, it does not show up in the simulator. This will seriously impact the survivability of fighter crews during missions.

Fidelity Quantification. Two possible solutions to quantify the difference between displayed and real performance are: (1) to develop a software test procedure whereby a simulated aircraft can automatically fly through a profile, measurements be taken and performance curves generated which can be compared to the secret performance charts in the intelligence manuals. (2) a short, written description for each threat platform be developed, detailing the assumptions, premises, simplifications, etcetera, used to develop the model, listing any known differences between model and real platform. This document could be classified above secret and available at an appropriate location for the pre- or postbriefing of appropriately cleared aircrew.

Sensitivity Analysis.

How Much Fidelity is Enough?

The bottom line for fidelity is that outcome reversal is unacceptable. Whatever kind of hostile engagement is involved, whether it be fighter vs fighter, or SAM vs fighter, or fighter countering a supersonic bomber equipped with supersonic missiles, the operational outcome of an engagement cannot be in error due to inaccuracies in the modeling. The standards used for comparison would be outcomes projected by intelligence data.

Additionally, each data set and algorithm must be individually accurate. This will allow one interface to be designed to load intelligence updates of all similar platforms without having to hand massage each data set or algorithm to make it work correctly. This requires that deficiencies resulting from one inaccurate data set or algorithm not be corrected by altering a different data set or algorithm.

Using Fighters for Illustration:

1. One example of this would be correcting a model's acceleration inaccuracies which actually were generated by an inadequate engine model, by alter-

ing airframe drag.

2. Detection modeling provides another example of the problem. Detection is a result of three functions: fighter radar or infrared sensor capability; atmospheric transmission in that particular frequency spectrum which is matched to the weather selected; and the signature or reflectivity of the target in that particular frequency spectrum. These three models combined should provide the resultant detection profile of the target by the fighter on that particular day. In this example, it should be obvious that the target signature model not be adjusted to compensate for problems in the sensor or atmospheric attenuation models.

Contractor Requirements

It still remains to be developed how to determine what accuracies are required in each algorithm and data set to insure that operational outcomes are accurate. This area, were it to be developed, would be especially useful in the writing of specifications for future modeling, and in the evaluation and acceptance testing which must be conducted as the part of each contract. This same need has been expressed by a foreign flag officer in charge of air simulator acquisition in his country. It should be obvious that the demand for this capability will increase dramatically as simulators increase in capability to the level where tactical development and evaluation of black programs can be thoroughly tested in simulators, and mission rehearsals can be conducted.

In conclusion we must:

1. Learn how to determine and specify degree of fidelity required for threat platform modeling.
2. Make our fleet crews aware of the specific fidelity limits in our current threat models.

Meteorology/Oceanography. The second significant area which must be accurately portrayed for advanced training is weather. The weather presented in current U.S. simulators is very simplistic. Essentially, it is limited to wind and reductions in visibility, often associated with altitude above the ground. This extremely limited, weather simulation has been prevelant for so long that modelers may have lost sight of the effects of weather on the crew and their weapons employment. When future, very capable ATS's are being designed, more complete, significant weather must be incorporated into the suite.

Background. Currently what is presented in visual trainers is area weather. In the F-14D ATS, if one dome is to be impacted by a cloud deck, then all domes must be subjected to the same cloud deck regardless of their separation from each other. This is grossly unrealistic.

In the real world, fighters geographically removed from each other's station are generally in different weather patterns. This is especially true in the scenarios seen today of U.S. Navy fighters being hundreds of miles apart, possibly on opposite sides of a front, yet in the same battle problem. The training shortfall associated with the simplistic appraoch to weather is that weather conditions are not loading up the crew with visual and weapon system tasks they will have in the real world. The success of the fighter interceptor crew of today is still very dependent on weather conditions.

Effects of Weather

Visual Contacts. The range at which a fighter crew can visually detect another aircraft is often directly dependent on weather. Weather will sometimes obscure certain aircraft at close ranges while permitting fighter crew to see other aircraft at long range; interestingly, it is not necessarily a factor that works equally on both participants. Color and light intensity, contrast of aircraft verses background, and visual clutter are examples of this inequality. In the real world, the crew must integrate a tremendous amount of visual information and reject visual clutter, of which weather is a significant part. In the real world of the fighter pilot, many contacts and friend/foe identifications are accomplished visually. There are numerous instances in my own experience flying F-14's that the first indication we had of even large aircraft being in our area was the visual sighting.

Influences Tactical Decisions. Pilots have historically used clouds and cloud formations to hide their aircraft while they attempt to increase their advantage in relative positioning. Weather will sometimes force tactical decisions. As an example: adverse weather is often something a fighter pilot will work to avoid, sometimes for safety, but also if he is trying to coordinate his flight visually with another crew in order to maintain electrical emission silence (EMCON) or trying to make a visual identification. Or, if his station is in an area that has significant weather, the crew is highly tasked visually, since an enemy weapons platform may be obscured by a set of clouds or haze until it is behind you. This is not where you want to pick up an enemy fighter coming toward you, or high speed bomber going the other way.

Additionally, cloud decks can be very visually disorienting in aerial combat. In real world training we are precluded from fighting in the vicinity of clouds. In combat, it should be obvious that we should know how to use clouds to our advantage. This means hiding in them, fighting through them, and learning how visually to fight an adversary using a cloudy environment to your advantage.

Missile Smoke. A blend of aerodynamics and meterological modeling is required to allow aircrew realistic, visual acquisition and to force real time decision making in ACM or in overland, power projection roles. The missiles which in real life produce smoke should emit smoke trails of the correct color, density, persistency, location and duration.

Infrared (IR) and Radar Contacts. With the new, long range infrared search and track sets coming into their own, the pilot may well choose to station himself either above or below a cloud deck which will have a tremendous impact on his detection of targets. We have also experienced numerous instances in which weather has ducted radar energy.

Clouds can present heat seeking missiles with a background clutter that is a function of sun angle, or hide an enemy from IR missile detection. The crew that has the most experience with this two-edged sword before going into combat will have a very real advantage. With today's safety constraints for training, this is training that can only be conducted in the simulator.

Modeling Types of Weather Systems. There are many weather systems which can be modeled. But all of them can be simplified to four types of building blocks: vertical cloud development, horizontal cloud development, reduced visibility in haze-type moisture and precipitation size, quantity and type. Haze and its effects come to mind when we think of the milk bowl of the South China Sea or the Mediterranean.

Ocean Surface. One other area of meteorology is its effect on the ocean surface. One of the design points for three of our visual systems is to ensure that the scene is adequate to permit daytime carrier landing practice, because the carrier landing environment is known to be challenging. As an example of this need, approximately one student per class at the FRS fails to qualify at the ship. Once in the fleet, crew are expected to make EMCON (all transmitters off)recoveries routinely during daylight hours, and they are expected to land on the first pass at the deck. An experienced crew has learned that they can improve their first pass using the waves, whitecaps, and the ship's wake to get a feel for wind speed, direction, and variability. Additionally, an idea of expected ship motion can be developed by looking at the ocean swells. Currently, the aircraft carrier motion is programmed into the visual scene, but the ocean surface is visually presented as a flat, blue surface with a generic texture applied to it which provides the crew no cues.

Conclusion. Weather effects are important enough and significant enough for the military and contractors jointly to expand the technical envelope as quickly as possible. The impact of the weather on each of the systems needs to be integrated and correlated. As an example you would not want to model the radar ducting effects of a low, inversion layer, and see a thunderstorm. To accurately learn to manage high, realistic workloads, and to make correct, tactical and weapons employment decisions, the fleet user must be able to train with ever increasing levels of fidelity in weather depiction.

Sun Effects. The last area to be discussed is the effects of the sun in the world of the fighter pilot as he trains for combat.

Technical Background. Where does the sun fit into the fighter pilot's world? For one thing it has brilliance. Currently, it is not possible to duplicate even some of the sun's brilliance in the daytime, visual environment presented in a dome. However, even today, it is possible to portray the sun's location, and it is possible to model many of the effects of the sun's brilliance on the visual, infrared and television sensors which are operational in the Navy now. It is probable that most future programs will be impacted by the sun's brilliance as well.

Tactical Background. Ever since WWI fighter pilots have used the sun to hide their aircraft from their adversaries. The sun not only obscures or hides a flight path as an aircraft crosses the small disk of the sun itself, but also there is an area surrounding the sun, which due to atmospheric conditions, will be sufficiently brilliant to prevent visual detection and preclude visual tracking. The affected pilot will try to minimize this trouble area by shadowing his eyes from the brightest sections using his hand. The precise size of this area is beyond the scope of this paper, but could be quantified into a reasonable spread of values which could either be selected manually by an instructor or, more preferably, automatically inserted into the battle environment based on the instructor-selected weather conditions. Targets which approach this area could fade and disappear as appropriate.

Effect of the Sun on Sensors. Another problem area presented by a brilliant sun is the direct effect on IR and visible light sensors. Ever since I've been flying fighters one of the tactics used to defeat an IR missile, or the adversary's ability to employ an IR missile is to put the sun in the background as the enemy pilot looks at you. For most IR missiles, the seeker will shift from the relatively weak IR source of an aircraft engine over to the higher energy source of the sun. This forces the downsun pilot to choose maneuvers which will minimize the likelihood of this happening. This training is hard to come by. Another effect is that television camera systems will shut down to protect the circuitry, or risk burning a track into the video signal receiver because of the energy intensity. If other IR or visual systems in the future are designed not to pick up the sun, the system will probably also process out any IR or visual targets within some angular distance of it.

Effect of Sun's Reflection. It's not only the sun itself. Any reflection of the sun will cause problems. For instance a low, setting or rising sun reflecting off the ocean will be a distraction for any visual or IR light system. Additionally, clouds can cause a reflection problem. Missiles can change lock onto these false targets, and pilots will be blinded to some degree by them. Therefore, it is necessary that not only the effects of the sun's brilliance be modeled, but also its reflection.

Shadows.

Shadows in the Cockpit. There are other reasons why multiple positions of the sun need to be presented. For the sake of brevity and lack of current technology, I will not detail the effects of sun and shadow in your own cockpit which significantly increase workload on the negative side, but provide significant motion cues on the positive side.

Terrain Shadows. The continually changing shadows cast by terrain and cultural features can either accentuate or hide key detail. The terrain changes visual aspect significantly as the sun changes in elevation. This is meaningful for all missions which are using physical geography for navigational reference, but especially critical in training for photo reconnaisance missions. Also, it provides the three dimensionality required for visual air-to-ground attacks. Once again, a pilot will be conscious of his background, e.g., if flying a low level route in the early morning or late afternoon, a fighter pilot will avoid flying in the sun if over shadowed ground, such as on the down sun side of a mountain. The pilot who does not have this situational awareness will show up like a star in a night sky when viewed from above.

Other Aircraft Position and Attitude. It is so obvious that it is often forgotten that airborne aircraft have shadowed surfaces. These shadows will, to a greater or lesser extent, be lightened slightly to none when the earth beneath the aircraft is illuminated by the sun from its zenith at noon time to a low elevation at sunrise or sunset. The shadows impact detection in that they can either serve to beacon an aircraft against a light background, or hide it, if the background is dark. The shadows are key attitude indicators in a visual fight. Experienced fighter pilots read these shadows in aerial combat for information as to the direction and attitude of the adversary. That this is of importance can be seen in the deceptive paint schemes which have been proven over the years in an effort to reduce the impact of shadows. There are two sources of shadows on aircraft: (1) On surfaces facing toward the sun, there are shadows cast by parts of the same aircraft. (2) On surfaces facing away from the sun, the entire area is shadowed.

A couple of general examples of the detection problem to which most people can relate: An aircraft or its contrail beaconing against a dark sky because the aircraft or contrail at altitude was caught in the light of the sun which has already set and is hidden from the earth-bound observer; or times when vehicular traffic was hidden by shadows from a driver who was looking into a sunrise or sunset.

Geographic Reference. The direct geographic reference is obvious when the sun is at a low angle. Of course there are those who will ignore the obvious. A couple examples come to mind: first of the west coast, Navy carrier pilot in the early 60's who, late in the afternoon, flew to the west when he needed to go east to find land. His excuse was that his gyro compass was unusable and that he couldn't read his standby compass because the sun was in his eyes. Also, during dogfight training one of the basics that is taught is for the crew to note the sun angle prior to the fight. That way, they have the only, overwater, directional reference in mind for the fight, departing the fight, and for debriefing.

Sun & Haze Combined Effects. Visibility on hazy days is a critical problem to the fighter pilot because it is a function of the position of the sun relative to your eye. The visibility when looking in the direction of the sun is much less than when looking in the direction opposite the sun. The greater the haze, the greater the difference based on direction of observation. Fighter pilots will take this into consideration when choosing their routes of flight. Also, if the sun is directly overhead, it will require substantially more aircraft energy and capability, and pilot talent to use the sun as a tool. If on the other hand the sun is low on the horizon, it requires much less aircraft energy and pilot talent to use the sun. An oblique or level maneuver could put the sun on your side. The piloting techniques are totally different. Reflections, shadows, maneuvering and visibility are all a function of the sun's location.

Conclusion. I believe it becomes obvious even from this incomplete list that the sun is a major driver of our fighter tactics. The sun and its effects must be modeled at multiple elevations in the sky. When low light level systems are considered, the moon must receive similar consideration.

Geographic Modeling.

Modeling of Real World. Accurate and realistic modeling of real world geographic features is required for crew training. Although to some people terrain modeling does not appear to be a major factor in the fighter world, this is not born out by combat.

1. Terrain masking of all sensors is used; even flat areas such as the plains of North Vietnam, or the deserts of the Mideast have been used successfully for masking aircraft from various weapons sensors.

2. Sophisticated, enemy and friendly overland tactics are almost always carefully planned around terrain features, if not for masking, then for orientation and timing. MIGCAP (offensive fighter)sweeps and TARCAP (defensive fighter) support of multidirectional power projection attacks are examples in which terrain is an extremely important reference.

3. Fighter performance and selection of maneuvers is constrained by the real world "hard deck" i.e. stone, mud, trees, etc. The visual scene must have the three dimensionality of the real world to give the ground rush, the masking, the feeling of closure and velocity of the real world, low altitude dogfight. Additionally, mountainous terrain in which the valleys and plains may have altitudes of 10,000 to 15,000 feet above sea level force aerodynamic and energy limitations on a crew as to the maneuvering tactics which can be used.

4. In order to plan and execute training missions such as escort and reconnaisance, the simulator generated terrain must be detailed to the point that it can be correlated to Joint Operation Graphic (JOG) charts (scale 1:250,000) and it cannot be generated by immediately noticeable polygons.

Modeling of Generic Terrain. In an effort to save money and yet gain substantial training benefit, it may be adequate for instructors to be able to generate a generic terrain which will provide most of the functional training a fighter crew requires even though it will not provide real world specific detail to work out rehearsal-type timing or ranging problems. An example would be training for defense of a constricted airspace such as the Strait of Hormuz between the Gulf of Oman and the Persian Gulf. A precise, geographical model might be too time consuming or too expensive to produce on short notice. The local simulator community should be able to construct a generic strait of water with generic islands, and a roughly sized and appropriately placed coastal area, city, mountain range and desert. The construction method of this geographic environment should be user friendly, through a graphic interface, and the use of a drawing program on a graphics workstation should be considered.

Scenario Generation

Background. A flight environment (including meteorology, ground threats, friendly, hostile and unknown aircraft with initial conditions and mission guidance, own aircraft initial conditions, and dome links) will be created by a person, probably other than a participating crew, and herein called an "instructor". Various triggers can be assigned to interactive CDP, and enemy performance can be regulated by assignment of "skill levels" to the simulated operators. The following are considerations raised with regard to the generation of this environment/scenario:

User Friendly. Technically the system should be user friendly to the extent that any FRS instructor could, through "on-line" Instructor Display System (IDS) screen helps, create or alter a scenario. The scenario, once completed should have levels of protection (passwords, etc.) to ensure that an altered scenario cannot be substituted for the original version without proper authority. However, the system should permit an instructor to save an altered version as a new document and file it with its own designator. To facilitate scenario development and review, recommend that the following be incorporated:

1. Menus from which the instructor can obtain Blue (friendly forces) and Orange (adversarial forces) platforms and their associated weapons and emitters.

2. A precision screen cursor driven by a trackball, mouse, joystick or touch screen interface with constant range/bearing and latitude/longitude readout for menu access and item placement into the geographic context.

3. A hard-copy form generated by the computer system on request which matches the IDS screen form to be filled in during scenario development. It should include blanks for the following: creator, unit identification; initial conditions; meteorology; geographic area; objectives; players (live vs computer); and controller organization. Meteorology should be in a format which would facilitate providing the aircrew a weather brief for their mission.

4. A verification program be available to check for missing or illogical data, and to direct the instructor to the location of the suspected error. To insure adequate flexibility to accommodate future scenarios and to overcome software errors, the instructor should be able to disregard the advice from the verification routine, and to accept a scenario with alleged errors.

5. A hard copy printout of the scenario developed should be available to the instructor, with a format similar to the original form used for scenario creation.

6. On-line help in pop-up or pull down menus to provide assistance for specific functions to the point of being suggestive, or directive.

7. During debug of scenarios the instructor must be able to preview the scenario at a workstation. This capability should include a computer aided "accelerated run through" capability.

Tools. A top-down approach should be employed for on-line scenario generation which allows the instructor to initially specify the generalities of the scenario and successively provide the details. For example, the first level page would provide a geographic display for force and meteorological conditions placement. The next level would be used to define specific friendly and raid formations. Then, for each formation, the aircraft type with weapon load-out, movement, etc., would be described. This implementation should provide page cueing methodology so that the instructor can navigate the page hierarchy expeditiously. This will facilitate design of the scenario.

For a work station on which to create scenarios, the instructor should be able to use an IDS screen on the TES, an unused IOS, or a debrief room console.

Targets in a scenario should be positionable by latitude/longitude, range and bearing relative to an instructor selectable spot or unit, or by eye on an IDS screen.

There should be editing software to enable the instructor to alter scenario characteristics, both during scenario generation, and during the actual training period itself.

Building scenarios will be research intensive to assure the proper selection of platform capabilities. Platforms should be pre-constructed with correct capabilities, tactics, and skill levels.

Macro, realistic packages should be pre-created which associate various platforms, variants, tactics, and skill levels.

Depending on scenario generation factors such as geographic scale and complexity of the threat package, the IDS screen could get cluttered to the point that it becomes unreadable. Controls should be available to the instructor to facilitate understanding of the data presented. At a minimum these should include zoom and changing symbology. The center of the zoom should be instructor designated, with both zoom in and zoom out directly selectable. Navy Tactical Data System (NTDS) symbology should be available to reduce clutter, and the Tactical Aircrew Combat Training System (TACTS) type of symbology available to increase information. This symbology change should be controlled by the instructor. One war gaming computer system uses color as a break-out tool and can display single categories of target individually, if required.

To provide for the total amount of information an instructor may require, there should be a capability for the instructor to designate a CDP or meteorologic feature and obtain its scenario data file which would include data such as preset speeds, track, altitudes, skill levels, fuel level, weapons load-out, triggers and objectives. A visual cue should correlate the item being investigated and its data block (e.g., the track symbol could brighten).

Regardless of the basic symbology used (NTDS, TACTS, etc), the instructor must be able to label each unit/ground site uniquely to be able to facilitate game construction and understanding. This identification must be displayable during the mission for situational awareness. Configuration of such labeling can be controlled adequately by a Standard Operating Procedure, and should have minimal technical constraints.

Meteorologic Initialization. The

instructor should be able to create weather patterns quickly. These patterns should be:

1. Developed graphically on the IDS screen using a pointing device and drawing/painting tools to designate areas and then to fill these areas in with appropriate conditions such as restrictions to visibility, cloud layers, temperature inversions, etc.

2. Capable of being filled with patterns and saved as a file which could be easily identified and retrieved for copy into new scenarios.

3. Capable of being stretched in any direction to facilitate tailoring weather for each scenario without being forced to start from scratch.

4. Designed with a definite gradient over a given area. Suggestions included loading two points on nearly opposite ends of the area and have the computer interpolate, or having the ability for the instructor to "paint" in a weather pattern over the geographic background.

5. Capable of being assigned a velocity.

Geographic Orientation. During set up, the applicable coast lines, airfields, targets, islands, latitude/longitude markers, range rings and radials, and other boundaries should be selectable for display. The instructor should be able to designate the center of the range rings and radials. These features should be suitably colored and shaded, and there should be the capability to zoom in or out to permit graphic IDS screen interface for unit and meteorologic feature placement.

To maximize training it is very desirable to provide an increased scope of geography beyond the local training area. The following are suggested:

1. World landmass, coastal outlines with generic land and water in appropriate places throughout any area in the world not specifically modeled to Defense Mapping Agency data.

2. The instructor should have the capability of defining any subareas of the generic landmass graphically on the IDS screen as being generic mountains, populated area, desert, snow, farmland, forest or minor water areas such as rivers and lakes.

Modeling Total Scenario. In certain, longer, training problems, the full spectrum of an Anti-Air Warfare (AAW) scenario should be modeled. This would include the inner air battle in which return to force (RTF) procedures could be established with the potential of Blue combatant ships and aircraft engaging F-14's not following RTF procedures, and battle damage to Blue ships caused by enemy missiles not successfully countered by the Blue, layered defense. This should also include Orange action against Blue aircraft other than F-14's. Examples would include high value units such as an E-2, or non-AAW Blue such as S-3's in the scenario.

Data Base Considerations. There is a firm requirement for the Fleet to have the ability to alter and create the threat/friendly data base at the local level for both FRS and advanced Tactical Development and Evaluation (TAC D&E) purposes. This includes both the geographic and the threat platform data bases. This is required to keep up with current intelligence regarding new capabilities, new platforms and revised assessments. Additionally, control must be provided to introduce threat tactics variables at the local level for training purposes. These might be developed to reflect intelligence of a threat platform observed infrequently practicing new tactics.

Generating Larger Scenarios. Currently, there are algorithms which model ACM. To increase the training value and "firepower" of the ATS, inclusion of a "smart" wingman (R2D2) for both Blue and Orange aircrew flying domes would be desirable. Computer generated voice calls from R2D2 should also be explored. Considering the normal paucity of information passed between cockpits in an ACM engagement, it is entirely possible that R2D2 could call his flight path and attack intentions. Larger scale coordination, ASW-27C training and visual lookout training would be greatly enhanced.

Rules of Engagement. (ROE) must be specified and consistently applied to the game. Blue vs Orange and Orange vs Blue ROE should receive conscious addressal and appropriate Blue and Orange weapon employment triggers should be available in the scenario creation format. These might include such triggers as a time delay for "weapons free" for one side after a first shot by the other side. Either a computer voice call, or an IDS screen cue is required for the Blue or Orange instructor correctly to change weapons release status of their domes or CDP.

Control

General. Control requirements in the ATS have been sorted into three categories: Blue, Orange and Master (or umpire) requiring the services of 1 to 12 instructors depending on the complexity and control detail of a simulator mission. Some control requirements are broad enough that they cover all three categories of control. These are presented in this section.

Tools. Familiarizing an instructor with a scenario. Instructors will have a difficult time determining the composition, structure, etc., of a scenario generated by another instructor. Instructors will require as many tools as possible to gain an understanding of a previously generated scenario:

1. Control of color brightness. When multiple contacts or various readouts overwrite each other on the IDS screen, the instructor should have a brightness control for each color to facilitate breaking out one side from another. This is especially important at the TES where the Master Controller will often have "truth" on his screen.

2. A summary hard copy with all scenario input data, and creation and modification dates. The format should match the generation and review format, and should be such that it can be printed onto standard, fanfold, letter-size paper for ease of filing.

3. A capability for the instructor to select a track or environmental feature and obtain its data file as discussed previously.

4. An "accelerated run through" capability as discussed previously.

Ease of Control.

1. An instructor should be able to set up quickly a simple Blue vs Orange engagement.

2. There should be a capability for an instructor to select a canned threat platform and modify its weapons load and other significant parameters, and place it into the mission while on-line.

3. During a run the instructor should be able to select a CDP and have the ability to maneuver it and change its significant parameters.

4. It would be valuable for the instructor to be able to select multiple CDP, group them, and then control them as a group, ungrouping them as required for more individual control.

Blue Control. As the complexity of scenarios increases, the ability of one instructor to control all forces and simultaneously monitor significant training objectives quickly decreases. Control of Blue forces will probably be split off to a separate instructor, or group of instructors, who will be referred to in this paper as Blue Control. This could be accomplished from a designated dome's IDS.

Instructor View. The control station for the fighters should approximate that of Air Intercept Controller (AIC) console with fidelity as to detection, data rates, data selection, and displays. Detection flexibility is required. Optimum would be selectable radar displays (e.g., APS-125, SPS-40, SPY-1). The intent is to give the fighters realism in their control. A side benefit would be to use rated air intercept controllers (AIC) and to bring them back into being part of the team by giving them realistic training with the fighters. With today's sophisticated and heavily protected threats, the full potential of the fighter/AIC team must be realized in order to survive. Additionally, the Blue Instructor should be able to select views from appropriate, individual units simulated as well as domes.

Battle Damage Assessment (BDA). Visual, radar and infrared cues of aircraft and ship damage is necessary for aircrew tactical decision making based on analysis and correlation of this real time data.

Orange (enemy) Control. For scenarios of low complexity and intensity, one instructor, the Master Controller (MC) at the TES, could probably be dual-roled as umpire, and as the controller of Orange forces. At a point of less than one tenth of the ATS capability, it will probably be necessary to split off Orange control functions from the MC. This separate instructor, or group of instructors will be referred to as Orange control. Orange control could be accomplished from a designated dome's IDS.

Orange Controller's View of the World. This needs to be selectable between "truth" and the picture gained from only the controller's sensor network. Ideally this should receive conscious addressal during scenario development and at game initiation. Since the goal of the ATS is training F-14 crews of varying skill levels, the amount of Orange sensor information should be selectable during play. For a one sided, standard FRS type trainer situation, the Orange instructor should probably be aided by ground truth.

Changing Options. Significant parameters and triggers are selected during scenario generation. The Orange instructor should be able to make manual on-line modification.

Cell Group/Ungroup. During scenario set up, to facilitate development, the instructor should be able to group individual aircraft with varying operator skill levels assigned into a cell and give that cell certain parameters and instructions for the mission. During the game, the Orange instructor should be able to break up a cell (ungroup) control the aircraft as individuals, and then regroup if appropriate.

Orange Radar and Radar Homing and Warning (RHAW). If a dome is assigned as a Orange aircraft, the crew should have appropriate, generic, Orange sensor information presented.

Orange Formation Changes. Incoming Orange bomber cells and fighter formations should use appropriate tactical doctrine associated with an aircraft's mission type and operator skill level. As an example: Tactical/threat simulation should provide capability for inbound bomber cells to change formations prior to attacking the objective. This should be modeled realistically, at least within expected bounds of aircraft aerodynamics.

Level of Control of simulated Orange systems countering a Blue power projection strike. The instructor should have adequate information to play the Commander of an Integrated Air Defense (IAD) network. Ideally, the instructor should also be able to play the commander of a Surface to Air Missile (SAM)/Anti-Aircraft Artillery (AAA) battery under his command.

Orange vs Blue Non-fighter and Neutral Units. The Orange instructor and platforms must be presented all the targets in a tactical environment (not just the fighters or high value ones) to force the controller to make realistic tactical decisions. An example of this would be the Blue deployment of decoys. The Orange commander should be able to select views from his individual units, simulated as well as domes.

Master Control. This function was derived after examining the functions and responsibilities of the instructor(s) operating the TES console. The instructor(s) at this station have unique requirements for data presentation and platform manipulation which will tax the simple layout of the TES. Ideally, the MC would be able to observe all decision makers' critical displays simultaneously. An example of the benefits from this capability would be to permit the MC to ascertain why a platform was not successfully engaged during the game.

Instructor Personnel. Flexibility in assignment of individual IOS's and the TES IOS is critical. The following are specifics which were raised:

1. Large scale problems may require all IOS's be manned (12 personnel) to monitor and/or control a mission. Each IOS should be able to communicate and control any dome.

2. Smaller scale, basic scenarios could be run by one or two instructors at the TES IOS with none at the individual IOS's.

TES Presentations. The TES IOS should have the capability of performing any function required to run the mission, and to call up any presentation which could have been called up on a dome IOS's IDS screen. (This does not include cockpit switch placement, but does include all cockpit displays such as the Heads Up Display (HUD), Multi-Functional Displays (MFD)'s, the Digital Display (DD) and the Tactical Information Display (TID). This also includes the views of "truth", and the Blue, and Orange instructor.

Event Tags/Flags. Automatic (inserted during scenario setup) and manual event flags should be available for insertion to aid in debrief. The flags should be unlimited in number and defined within certain classes so that during debrief a particular class of flags can be called up. For example one class of flag could be Blue weapon expenditure. The instructor's voice should be recorded in memory and capable of being played back so that the reason the flag was inserted can be quickly retrieved and accurately ascertained during debrief or mission evaluation.

New Player Insertion. The MC must have the capability to insert into a mission that is running, platforms that were not originally scripted into the scenario.

Trigger Actuation Signals. As the triggers of Orange or Blue CDP are actuated, a visual signal should be displayed to controllers at all appropriate IOS's. The instructor should be able to hook the trigger to obtain the trigger criteria and triggered action. This data should also be available in hard copy for debrief. The voice annotation capability discussed previously should be available here as well.

Tailoring Platforms. During a mission, while on line, the MC should have the ability via graphic interface and menus to tailor Orange or Blue characteristics, or parameters associated with the platform, including skill level, triggers, emitters, etc. It is not required to change emitter parameters on line such as pulse repetition frequency or pulse width.

To facilitate Orange and Blue interaction, and to minimize time lost running through sections of a game that are not germane, it is required that the MC have the capability to slow down or speed up a scenario, or at least to step to a particular time after scenario start.

Data Extraction. A mission is virtually wasted if it cannot be debriefed and lessons learned. The primary method for student debrief and mission analysis is real-time playback of the mission. Playback of individual cockpit displays as well as any level of control should be provided. Means must also be available to debrief and analyze selected parts of a mission on standard Navy computers in order to avoid scheduling conflicts in the ATS.

Scaled Symbology. With large areas portrayed on the IOS IDS screen, it may be necessary to simplify the symbology presented. However, it is also necessary to be able to see accurate cell or group composition when the area is zoomed in upon. The instructor should be able to change IDS screen symbology between NTDS-type and TACTS-type. A capability for an instructor to assign different colors to aircraft by type within a cell would also facilitate understanding of a complex or cluttered IDS screen.

Required Data Elements. Another debrief/analysis tool is a detailed readout of mission events. Elements for extraction should be selectable via menus to tailor the extraction to mission requirements. This information should either be displayed on the IDS screen or printed for subsequent analysis. The following is a suggested data base:

1. Aircraft tracks and positions.
2. Selectable time ticks on platform tracks -30 second ticks with 5 minute annotations.
3. Range, bearing, aspect, closure, altitude difference between selected pairs of aircraft.
4. Position, heading, altitude, g and weapon systems status for selected (hooked) aircraft.
5. Communications.
   - Record individual communication channels with timing tracks.
   - Provide capability to replay with and without jamming.
6. Sensor information.
   - Sensor page for selected platform contacts, strobes, and data link target displayed on x, y plot. Ground truth should be able to be superimposed on sensor information.
   - Current Electronic Counter-Counter Measures (ECCM)/weapon system switch positions (on/off, gain, mode, scan volume, etc).
   - Sensor detections/lost detections summary page.
   - Composite: Big picture with all strobes, contact, data link targets. Ground truth should be superimposed as desired.
7. Missile Information.
   - Display weapon status page for selected platform--LAR, weapon selected, radar mode, Electronic Counter Measures (ECM) observed.
   - Actual target ECM. Launch parameters for each missile-time, range, aspect, closure, altitude difference, weapon system parameters, actual ECM, flyout result (time, miss distance, fuze, result).
   - Composite: Summary page of all missile shots, launch times, ranges, aspect, Vc, altitudes, flyout times and results.
   - Missile shots superimposed on track x, y, z plot to correlate track history with missile employment.

Records for Analysis. Mission analysis requires extensive mission playback which, if conducted on the ATS, could present considerable ATS scheduling conflicts. It is highly desirable to have the capability to download appropriate mission data to a standard, Navy, IBM PC AT compatible computer workstation (e.g., Z-248) for offline analysis, which will also make the results of the simulator sessions more readily available to instructors and squadrons for review and analysis. The following data should be provided: Voice; time; ground tracks with NTDS symbology; 3-D tracks with TACTS format; event tags; trigger markers; instructor flags on tracks; and crew presentation(s) either running or as stripped, at least surrounding the above flags/tags/triggers.

Audio/Video Recordings. Each channel of audio should be recorded on a standard media, which can provide selectable voice playback. Additionally, cockpit communications should be recorded on the video tape recording of cockpit displays at individual domes. This should be in a format or changeable to a format which could be used at the squadron level.

Flags. At the debrief station, the instructor should get a listing of flags in chronological order. The information should be available on the IDS screen and in hard copy. The format should be directly useful for debrief, and allow the instructor to access the specific event based on time for an appropriate rerun. The method by which the instructor inserts flags should automatically permit voice annotation.

Location of Kills. The location of kills should be recorded and displayed for debriefing geometry and tactical effectiveness. They should appear on the MC's picture of "truth". The aircraft killed should freeze in position and alter visually (e.g. the color or symbol, or lighten the symbol in shade) allowing the debriefer to see and debrief the positions of kills.

Video Recording of Debrief. The debrief facility should be equipped with a video camera to record the debrief interplay between crews, instructors and debriefer. Many invaluable comments are made through short term recall during the debrief that would facilitate analysis of a mission. This is true from basic airwork and operational procedures training through complex, multi-dome missions.

Miscellaneous.

Insertion and Removal of Trainees. It is imperative that the ATS have the capability for domes to be inserted into and removed from a mission to maximize training without freezing the game or losing a track file. The dome should be able to replace any target in the scenario. This will allow the MC to insert a dome into a pending engagement (beyond sensor range) complete the engagement, and be able to move the dome crew to another location for another engagement.

Modeling of Communications Jammings. Voice communications and data link jamming must be simulated intelligently. It must be based on power and relative geometry of jammer and communicating transmitters and receivers.

Security. The ATS is currently being developed for operation at the secret level. There are taskings which will require higher levels of security than these trainers are designed to accommodate. Research must be initiated on security requirements for operation, and retention/destruction/storing of magnetic media above secret, possibly including Special Compartmented Intelligence (SCI) scenarios. Emissions and building physical security should be considered. This is another reason why the system should be sufficiently user friendly for military aircrew to operate. Although Contractor Operation and Maintenance of Simulator (COMS) and Contractor Simulator Instructors (CSI) personnel may have clearances to access classified information, they probably will not meet the "need to know" briefin criteria.

Software Updates. As the software operating system is revised, it should always be capable of running old scenarios, or conversion programs must be concurrently developed.

High Energy Weapons. The use of range finders and energy weapons can be anticipated and should be modeled from appropriate platforms. Suitable laser affects should be modeled in the simulator. Weapon seeker destruction, and temporary crew blinding could be simulated if these are appropriate responses to being lased. This issue should be researched and modeled as appropriate.

Debrief Facility Capacity. Debriefing missions that include five cockpits requires a facility capable of debriefing an entire squadron, not just the active participants. Additionally, the facility should allow instructors (up to 12), analysts, and observers as well as participants. Auditorium-style seating for 50 persons would not be excessive.

Linking with Other Trainers. Currently F-14D/E-2/A-6/F-14A (2F112)/F-18 aircrew trainers can not be linked in a common, concurrent, simulated tactical environment. Major limiting factors are (1) Trainer to trainer communication (e.g. Miramar to Whidbey; Miramar to Oceana) (2) Compatibility of trainer host computation systems (hardware and software), and (3) Correlation of individual trainer data bases. At least at the system level, F-14D/F-14A (2F112)/A-6G/A-6SWIP/E-2/FA-18 aircrew trainers computation systems will be compatible in that the new trainers will employ the same computers as will the rehosts for the E-2 and F/A-18. Priorities should be established for linking.

Summary

We must be able to use the ATS to train for complex battle problems. We cannot be satisfied with functional training of one or two crews. As we develop weapons whose technical operation we want to protect, as we see threat countries capable of fielding more sophisticated weapons systems, as we force the intelligence community to provide us with accurate threat models, the simulator will be THE place for fighter practice against advanced enemy weapons systems.

Done correctly, we will be able to:

--Fight against threat systems in the quality, quantity and mixture we expect to see in combat.
--And practice employing our latest weapons, tactics and sensor systems in a realistic environment.

I would suggest that our first, and immediate priority needs to be letting our crews know just what it is they are fighting against in our ACM trainers. After this we should maximize our threat fidelity. Then, blend improvement in geographic sun effects, meteorology and ocean surface. We must think of our simulators as operating in a sensor environment. We should incorporate what is currently available in these areas as soon as possible. We must communicate the fleet's interest in modeling parts of the visual environment which are not currently available. And, we need to closely monitor the feasibility of future improvements for all fighter simulators as technology advances.

Glossary of Terms and Acronyms

AAA--Anti-Aircraft Artillery
AAW--Anti-Air Warfare
ACM--Air Combat Maneuvering
AIC--Air Intercept Controller
ATS--Aircrew Training Suite
ATF--Advanced Tactical Fighter
Blue--(friendly forces)
BDA--Battle Damage Assessment
CDP--Computer Driven Platform
COMS--Contractor Operation and Maintenance of Simulator
CSI--Contractor Simulator Instructors
DD--Digital Display
ECM--Electronic Counter Measures
ECCM--Electronic Counter-Counter Measure
EMCON--Electronic Emission Control
FIT--Fleet Introduction Team
FRS--Fleet Replacement Squadron
HUD--Heads Up Display
IAD--Integrated Air Defense
IDS--Instructor Display System
IOS--Instructor Operating Station
JOG--Joint Operation Graphic
MC--Master Controller
MFD--Multifunction Display
MFT--Mission Flight Trainer
NTDS--Navy Tactical Data System
NTSC--Naval Training Systems Center
ORANGE--(adversarial forces)
R2D2--"Smart" Wingman, a CDP that cues off the leader and the enemy
RHAW--Radar Homing and Warning
ROE--Rules of Engagement
RIO--Radar Intercept Officer
RTF--Return to Force
SAM--Surface to Air Missile
SCI--Special Compartmented Intelligence
TAC D&E--Tactical Development and Evaluation
TACTS--Tactical Aircrew Combat Training System
TES--Tactical Environment System
TID--Tactical Information Display
WST--Weapons System Trainer

89-3304-CP

AN EXPERIMENTER OPERATOR STATION FOR HELICOPTER FLIGHT SIMULATOR RESEARCH AND TRAINING

Thomas A. Kaye*
Bell Helicopter Textron
Fort Worth, TX

and

L. Michael Freeman**
The University of Alabama
Tuscaloosa, AL

Abstract

The increasing use of flight simulators for pilot training is primarily driven by economic and safety considerations. A very labor intensive research effort is usually required in order to verify the basic assumption that the skills a pilot develops in the simulator are the same as the skills required to fly the actual aircraft. Studies that demonstrate a positive transfer of learning to the pilot typically require human factors researchers to perform many repetitious and tedious tasks, particularly in the area of data acquisition and statistical analysis.

The Experimenter Operator Station (EOS) was specifically designed as a multipurpose, research-oriented, operator support station used to control a low-cost helicopter flight simulator while simultaneously monitoring and recording pilot performance data for subsequent replay or statistical post-processing. The system is a mouse-oriented, menu-driven, interactive program written in the C language for the UNIX-based Silicon Graphics IRIS 2400 Turbo workstation. Structured program design methods were used to create a modular software package capable of handling data acquisition and graphic display of flight performance variables at real-time speeds of up to 30 hertz as well as intermittent user-requested simulator control directives. Recorded data files can be created during simulated flights for subsequent statistical analysis or flight replay and are intended to provide human factors personnel of the U.S. Army Research Institute with insights into optimal methods for simulator pilot training.

Introduction

One of the more visible trends in the training community today is the growing use of flight simulators for pilot training. Simulators are being used to train pilots in everything from basic flying skills to management of the complex array of systems associated with the current generation of advanced aircraft. Several appealing reasons which support the use of simulation for pilot training [1] are:

1. increased efficiency, as training will not be obstructed or restricted by factors such as adverse weather conditions, airspace limitations, or aircraft availability;
2. increased safety during the training period;
3. lower overall training costs;
4. the facility to practice situations which for reasons of expense, safety, and practicality cannot be rehearsed under actual flight conditions; and
5. the reduction in operational and environmental disturbances.

In short, the motivation for using training simulators is to provide greater training effectiveness at a reasonable cost. A careful trade-off study should be performed before, during, and after the development of a flight training simulator to determine the most effective way to build and use the device. Such a study involves comparisons between various factors resulting from simulator versus actual aircraft training on a maneuver by maneuver basis. Factors which must be examined include:

1. time required to achieve proficiency for a given maneuver,
2. level of proficiency attainable, and
3. total cost to achieve proficiency.

*Computer Engineer, Bell Helicopter Textron
Member AIAA

**Assistant Professor, Aerospace Engineering
Senior Member AIAA

A trade-off study is especially important if the prototype flight simulator is to be replicated many times for large-scale pilot training; however, performing such a study can be tedious, time-consuming for the experimenter, and expensive to complete. In an attempt to ease experimenter workload and ultimately reduce the time and cost associated with these kinds of studies, the idea for an Experimenter Operator Station was conceived.

Motivation for Experimenter Operator Station Development

One of the primary reasons for development of the UH-1TRS (UH-1 Training Research Simulator) was to provide the U.S. Army Research Institute's experimental research pyschologists with the means to study the effectiveness of low-cost helicopter simulator pilot training techniques. These human factors researchers are interested in knowing how well pilot trainees can learn to fly various primary training maneuvers in the simulator, and to what degree this training will affect their performance of those maneuvers in the actual UH-1 helicopter. They are also interested in determining which features of the UH-1TRS are, or are not, important for pilot training and which features will, or will not, need to be enhanced.

Initial efforts by personnel of The University of Alabama Flight Dynamics Laboratory (UAFDL) were focused primarily on the development of a low cost helicopter flight simulator which would accurately reproduce the visual and motion cues experienced by pilots during actual helicopter flight. The resulting simulator has been flown by numerous experienced UH-1 pilots who have verified that its handling qualities are indeed very much like those of the actual helicopter. The UH-1TRS was also analyzed mathematically to verify that the dynamic response of the simulator to pilot input was the same as the response of the actual helicopter over the range of flight conditions in which training would occur [2].

Although the engineers and UH-1 pilots are satisfied that the simulator's response to pilot inputs matches that of the actual UH-1 helicopter, the Army will be satisfied only after studies demonstrate that simulator-trained pilots are actually able to learn the skills required to fly a real UH-1 helicopter. Experiments of this type involve sample populations of pilot trainees and require that the human factors researchers perform many repetitious and tedious tasks especially in the area of data acquisition and statistical analysis. Thus, the need exists for a multi-purpose, research-oriented, operator-controlled station that will assist the human factors researchers with their various experiments. To assist the simulator operator (or instructor), such a system should be able to:

1. command and control all simulator functions through a menu-driven graphical user interface,
2. monitor pilot performance in real-time during training using graphical displays of various flight parameters versus time (or position), and
3. record and playback flights for debriefing purposes.

For the experimenter, the system should also be able to:

4. monitor pilot performance in real-time during experiments while automatically and/or manually recording important events associated with the current experiment and
5. record and playback flight data for more detailed post-flight analysis using statistical methods.

Additionally the system should be:

6. relatively inexpensive, using off-the-shelf hardware and software where possible, and
7. expandable to allow for future enhancements of its capabilities.

The Experimenter Operator Station is specifically designed to fulfill these needs.

Description of Experimenter Operator Station

The Experimenter Operator Station (EOS) is a menu-driven, multi-functional, workstation software package designed to provide user-friendly command and control of the UH-1TRS helicopter flight simulator system. It has been developed in the C language using the UNIX-based Silicon Graphics IRIS 2400 TURBO workstation as the host processor. The EOS is intended to serve as the control center for the UH-1TRS as well as other flight simulators in the UAFDL. The powerful graphics capabilities of the IRIS make it an especially well suited platform for real-time monitoring and interactive control of flight simulators.

This paper provides a brief outline of the Experimenter Operator Station's design and future possibilities. It first outlines the system requirements for the EOS, describing each class of functions along with the inherent difficulties associated with their inclusion into the EOS system. Next, both the hardware and software configurations are described as is integration of the EOS into the UH-1TRS system. Special consideration is given to software optimization techniques that are used to improve real-time performance. Finally, conclusions are presented with recommendations for future enhancements to the EOS package.

System Design Requirements

The Experimenter Operator Station satisfies a minimum set of requirements which were stipulated by the Army Research Institute (ARI) to ensure that its research needs will be met. Beyond the explicitly stipulated ARI requirements, the EOS design team formulated a number of additional, self-imposed requirements which reflect both the future goals of the Flight Dynamics laboratory and good software development practices.

Classification of EOS Requirements

Each of the requirements fall into one of four major categories:

1. command and control of simulator functions,

2. monitoring of flight parameters and pilot inputs,

3. utilities, or

4. general requirements.

1. Command and Control Requirements

The UH-1TRS is a complex device consisting of a number of special purpose computers operating both individually and cooperatively to share the heavy processing requirements associated with flight simulation. Although each computer is programmed to operate in a relatively autonomous fashion, it is still necessary to direct the entire cluster of computers to collectively execute various simulator functions and tasks. These tasks range in complexity from the very basic (start and stop functions) to the very complex (flight recording and data replay). The mere fact that the simulator system can perform these functions or tasks is not enough. What is required is the means by which a human operator can direct the system to perform these tasks and then verify that the task has correctly been executed. This is in essence the meaning of command and control. The EOS is expected to not only initiate commands to the system but also to recognize commands that have been issued from the cockpit (by the student or instructor pilot) and to respond with appropriate feedback messages on the screen.

There are three separate Command and Control modes which are required by the UH-1TRS system. They are the Run/Record, Flight Replay, and Data Replay modes.

Run/Record Mode

The Run/Record mode permits an operator to initiate and terminate live man-in-the-loop simulations, record pilot inputs and/or flight data, freeze the system at any point in a simulation, reset the system to some user-defined set of initial conditions, modify an existing set or create a new set of initial conditions, and view an instant replay of the last two minutes of flight, resuming live control from any point in the replay segment.

Data Recording

The collection of data is paramount to any successful experiment and must be given proper consideration. With state of the art, high speed digital computers, so much data can be generated in such a short time that memory limitations can easily be exceeded if careful attention is not given to the process of data acquisition. The UH-1TRS host computers simulate flight by repetitively integrating a set of differential equations which describe the behavior of the helicopter with respect to time. These equations take into account external inputs caused by such factors as atmospheric wind, turbulence, and air temperature as well as control inputs caused by the pilot himself. At least thirty times per second, the equations are solved and a discrete frame of data (analogous to the frames of still photos that make up a motion picture film) is generated. Included in these data is a series of 22 variables (56 bytes) that are associated with the state of the aircraft during that thirtieth of a second. Thus, the EOS must be able to record a maximum of 100,800 bytes of data for every minute of flight time. Recording this magnitude of data for every experiment can be wasteful of both memory and time since the more data recorded, the more difficult the task of reducing it for later analysis. To alleviate this problem, the EOS is designed to permit the experimenter to select from a range of recording frequencies to match the specific needs of a particular experiment.

Flight Replay Mode

Flight Replay Mode permits an operator to replay a flight by supplying the simulator system with previously recorded pilot inputs from a data file rather than live pilot inputs through the cockpit controls. The flight replay may be started, stopped, reset to the beginning, paused and continued. In addition, a live pilot can "fly-out" of a recording at any point (which automatically puts him back in Run/Record mode), or record flight data for closer analysis using the

EOS. Functions that are associated with the Flight Replay mode are described below.

Data Replay Mode

In Data Replay mode, the operator or experimenter has the capability to replay flight performance data files which, unlike the pilot input files used in the Flight Replay mode, do not require the entire simulation system to be used in order to recreate the flight. Instead, the EOS alone is used to view the replayed performance data. Since no new data are actually being generated in this mode (only retrieved), the data may be played back at faster or slower than real-time speeds. This is helpful for quickly locating an interesting point in a flight by scanning through the data rather quickly. When the desired point has been located, the playback rate can be slowed, stopped, or even reversed for more detailed analysis of the event. Functions that are associated with the Data Replay Mode are described below.

2. Monitoring Requirements

Another very important function of the EOS is to provide feedback to keep the operator informed about all aspects of the simulator session. Information about the status of the EOS system, simulated environment conditions, and pilot performance are all required if adequate control of the UH-1TRS is to be maintained. Monitoring is also required for analysis of the experimental data being generated. The analysis may either occur during a live simulation or after a flight session using flight and data replay functions.

Data Plots

The most important use for EOS monitoring is to provide detailed feedback about pilot performance. Recording of data parameters is generally done for the sole purpose of determining how well the pilot can fly an aircraft. Statistical post-processing of these data provides a mathematical way of determining pilot performance; however, these methods hide the actual appearance of the raw data and do not provide the kind of immediate feedback required to detect subtle pilot errors. Real-time monitoring of graphical representations of pilot performance plots is the best way to acquire a general understanding of what is occurring during a live flight or to analyze in detail what happened during a previously recorded flight. Typical plots would display Airspeed vs. Time, Altitude vs. Time, Any Control vs. Time, Latitude vs. Longitude (Horizontal Situation or Groundtrack), and Altitude vs. Distance from Landing Zone (Glideslope).

3. Utilities Requirements

The purpose of utility functions is to provide various tools for enhancing the effectiveness of the EOS or for reducing the operator's workload. Many of the utilities are similar to database functions and provide access to volumes of information stored on disk. The utility functions include:

1. a pilot biographical database,
2. screen display options (color, plot arrangement, etc...),
3. statistical analysis tools (provided by off-the-shelf software), and
4. a data file management facility to eliminate the need for a user to be conversant with the UNIX operating system.

4. General Requirements

There are three self-imposed requirements that were stipulated by the EOS design team in an attempt to maximize the potential of the final system while at the same time minimizing development problems. They are:

1. compatibility with both the UH-1TRS as well as UAFDL simulators,
2. modular programming to ease multiple programmer development conflicts and to promote software maintainability, and
3. design for future expansion.

Aspects Of Hardware Design

Two helicopter flight simulators were produced as a result of the UH-1TRS development project. The first is a developmental prototype called the UAFDL Helicopter Flight Simulator which is located in the Department of Aerospace Engineering at The University of Alabama. The second is the UH-1TRS belonging to ARI and is located at Fort Rucker, Alabama [3]. There is a noticeable difference between the configurations of the two simulators (primarily with respect to their visual systems) which the EOS hardware design must take into account so that it can be effectively used with either simulator configuration.

Hardware Design Objectives

Four major objectives for selecting EOS hardware components are to:

1. satisfy all system design requirements

2. ensure EOS compatibility between UH-1TRS and UAFDL simulators,

3. keep design simple to minimize development time and hardware costs, and

4. use only existing simulation hardware if possible.

Simulator Configurations

The UH-1TRS was built around a Singer-Link 2B24 instrument and procedural trainer that lacked out-the-window visuals altogether and used a very simple helicopter math model [2]. The 2B24 cockpit and motion base system have been retained but the math model, visual system, and all computers required to operate the UH-1TRS are new equipment, designed, developed, and installed by a team of Electrical and Aerospace Engineering faculty and graduate students at The University of Alabama. In order to facilitate hardware integration and software development for the UH-1TRS, the UAFDL Helicopter Flight Simulator was first constructed in The University of Alabama Flight Dynamics Laboratory. All software is developed and tested on this prototype before installation on the UH-1TRS.

The hardware configuration for the UAFDL Helicopter Flight Simulator is shown in Fig. 1 and consists of a UH-1 cockpit cab, a Heurikon HK68 Computer (also referred to as the Control Input Processor or CIP), a Floating Point Systems FPS 5305 Array Processor (AP), a Digital Equipment Corporation DEC MicroVax II, two Silicon Graphics IRIS 2400 Turbo workstations, and two mirror-beamsplitter optical units. The five special purpose computers are all linked together with a high speed Multibus, which is a custom design by The University of Alabama Computer Design Laboratory [4]. The pilot controls are located in the UH-1 cockpit and are read by the HK68 via Analog-to-Digital and Digital Input channels. The MicroVax II acts as the host for the FPS 5305 Array Processor which is responsible for handling all flight dynamics computations based on the UH-1 math model. The AP writes the resulting position and orientation vector to Dual Port Memory located in the HK68. The Dual Port Memory is accessible to both IRIS 2400 Turbos which act as front and side window Computer Image Generators (CIGs). The position and orientation vector is used by each CIG to create a high-resolution, three-dimensional perspective, out-the-window image which the pilot views from the UH-1 cab through the mirror-beamsplitter infinite focal length optical units. Alternatively, the simulator can be configured to use one IRIS to run the EOS software while the other provides front window visuals.

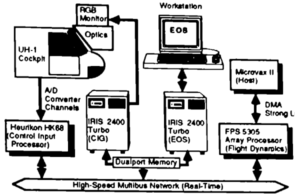


Fig. 1 UAFDL Hardware Configuration (with EOS)

The hardware configuration for the upgraded UH-1TRS is shown in Fig 2. The major difference between this configuration and that of the UAFDL simulator is the addition of a motion base system and the replacement of both IRIS 2400 Turbos by two new Delta Graphics Computer Image Generators (CIGs). The new Delta Graphics systems differ from the IRIS Turbos in that they operate synchronously at a constant 30 frames per second rate whereas the IRIS Turbos generate images asynchronously at a rate anywhere from fifteen to twenty frames per second. In addition, the Delta Graphics CIGs are not stand alone units like the IRIS Turbos but rather, must be hosted by the MicroVax II (which also plays host to the FPS 5305 Array Processor). The IRIS systems used only the position and orientation vector from the AP via the Dual Port Memory to compute each new image but, the Delta Graphics CIGs require the AP to compute, assemble, and send to each unit a rather large (1 Kilobyte) data packet via the MicroVax and DR-11W data bus at a synchronized rate of 30 hertz. This additional computational load adds a great deal to the frame times of both the AP and MicroVax which limits the reserve computational capacity available for use by the EOS system.

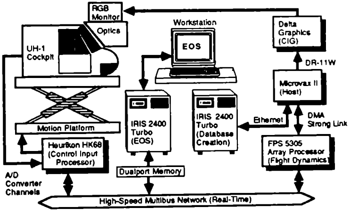


Fig. 2 UH-1TRS Final Hardware Configuration (with EOS)

EOS Hardware Components

Since the upgraded UH-1TRS configuration does not require either of the IRIS workstations, one of them has been developed into the Experimenter Operator

Station. Use of the IRIS satisfies nearly all of the previously identified Hardware Design Objectives and does so for both simulator configurations. The IRIS is the most logical choice of computer to host the EOS because it is an existing piece of hardware which is not required for the visual system upgrade. It is already installed and set up to receive output flight parameters over the Dual Port Memory and it also uses the Dual Port Memory to transmit data (commands) back to the AP. With its UNIX operating system and powerful C compiler the IRIS is an excellent software development platform. Its ability to produce high-resolution color graphics has been well demonstrated as has its suitability as an interactive workstation. It allows up to eight megabytes of Random Access Memory (RAM) which can be used for real-time storage of up to 30 minutes of continuous data recording time and has a forty megabyte hard disk drive with tape backup capability to permanently archive large data files. Details of the Dual Port Memory and Multibus may be found in Reference [4].

Aspects Of Software Design

Despite the fact that the EOS relies heavily on the computational power of the IRIS 2400 Turbo workstation and specialized communications hardware, the most important factor determining the effectiveness of the EOS is the quality of its software. Hardware represents the ultimate potential of a system but software determines the degree to which that potential is reached. The extensive list of EOS requirements makes it clear that a tremendous amount of software is needed in order to provide the system capabilities previously outlined. The difficulty involved with programming the many desired functions for the EOS is accentuated even more by requirements for interactivity and real-time performance. Under these conditions, only a well thought out development strategy employing structured programming techniques can yield satisfactory results within a reasonable length of time.

Software Design Objectives

Of paramount importance to any software development effort is a clear project specification which defines objectives, priorities, and a practical implementation plan.

The objectives (in priority order) for the EOS are:

1. Design a solid program framework to which all function modules can be added and from which they can be executed. Ensure that this architecture leaves sufficient room for expansion and will not prohibit inclusion of desired future capabilities.

2. Develop an efficient user interface that requires minimal user effort and CPU time for function selection. This includes designing an interactive menu system and functional screen layout. Every effort should be made to maximize the use of the capabilities of the IRIS hardware.

3. Program essential algorithms for writing real-time data to and reading it from other computers in the UH-1TRS system. Data communications and Input/Output are absolutely critical functions of the EOS. Since data transfer must take place at a maximum frequency of 30 hertz, these algorithms must be highly efficient to allow enough time within each thirtieth of a second frame for execution of Command, Control, and Monitoring functions,

4. Develop and test required Command and Control functions. Priority is placed on programming only the essential functions required to run, record, and replay flights and output data. Special features associated with data replay have been considered in the design but reserved for development in a future version of the EOS.

5. Develop and test essential monitoring functions. Priority is placed on creating a selectable set of pre-defined data plots and a basic status panel display. Special features such as a Tactical Map and Instrument Repeaters may be added later.

6. Develop and test the most useful Utility functions. Many other convenience utilities will be added after an initial version of the EOS is operative. To save development time, Statistical and Data File Management will be handled by a commercially available statistical analysis package. Priority should be placed on the Pilot Biography Utility since it is required to generate identifying labels for recorded data files.

Overview of Software Design

The EOS is a highly interactive program intended to perform many autonomous functions (such as data recording and plot display) while simultaneously listening and responding to specific user requests. The difficulty associated with such a task stems from the fact that autonomous functions are well-defined events which take place at very regular intervals while user requests are typically made at very irregular intervals and are unpredictable in terms of the processing time they require. Since neither user requests nor autonomous functions may be given the undivided attention of the central processing unit (CPU), the EOS must be programmed to handle both of these contrasting demands without allowing either demand to interfere with the other. The style of programming used in this type of scheduling situation is known as Event Loop Programming. A vast majority of computer programming is sequential in nature, with execution of the program code proceeding from start to finish along a well-defined execution path. In contrast, an event driven program continually cycles through a loop until an event occurs (internally generated or user requested). It responds to the event by executing the appropriate procedure, and then returns to the loop when execution of the procedure is complete, ready to process the next event. Since the type and order of user requests is arbitrary the resulting execution path is also arbitrary and therefore the program must allow for (or guard against) all possible user selections. The basic concept of Event Loop Programming is illustrated in Fig. 3.

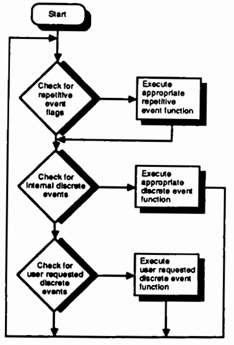

Start
Check for repetitive event flags
Execute appropriate repetitive event function
Check for internal discrete events
Execute appropriate discrete event function
Check for user requested discrete events
Execute user requested discrete event function

Fig. 3 Typical Event Loop Program

Three different types of events can request execution of two different types of procedures in a typical event driven program. The three event types are: repetitive events, internally requested discrete events, and user requested discrete events. Repetitive events initiate execution of a repetitive procedure. For example, a variable called "record_flag" could be used in a loop to indicate that data are to be recorded for each pass through the loop that the flag is TRUE.

Both internally requested and user-requested events initiate execution of discrete procedures which are executed only once for each request. An example of a discrete procedure initiated by an internally requested event would be automatic scaling of an on-screen plot if the value of a parameter is out of the visible plot range. An example of a discrete procedure initiated by a user requested event would be terminating the program to return to the operating system. Note that a discrete procedure is often used to turn on or off a flag that controls a repetitive procedure.

By utilizing the method of event loop programming, the larger task of programming the entire EOS system is immediately broken down into three smaller tasks, namely:

1. develop an event loop which will provide the proper branching to appropriate functions based on various types of events,

2. develop a method of generating requests (i.e. a User Interface), and

3. develop appropriate discrete and repetitive procedures which satisfy the requirements for Command and Control, Monitoring, and Utility functions.

A major advantage of this type of software architecture is that the program is easily expandable. Once an event loop and user interface are created, it is very easy to add additional functions to the system in a modular fashion.

Main Event Loop

After initialization of the IRIS hardware, data structures, and various default configuration files, the program enters the Main Event Loop where it cycles until an event is detected. Based on the nature of the request, the program will either execute a function, enter the Real-Time Simulation Event Loop, or terminate. The Real-Time Simulation Event Loop is a loop that has been optimized for handling real-time operations of the EOS which must occur while the simulator is being flown and data are being recorded. All procedures which require more execution time than the allowable thirtieth of a second are rendered inaccessible from the real-time loop. Whenever the simulator is not being flown, the EOS operates from the Main Event Loop which allows access to all functions regardless of their execution times. Fig. 4 shows a flow chart of the Main Event Loop. Note the pause at the top of the loop while the program waits for a 60 hertz synchronization signal. This "sync" signal (which the EOS uses for timing purposes) is generated by the

IRIS hardware at the end of each screen refresh cycle.

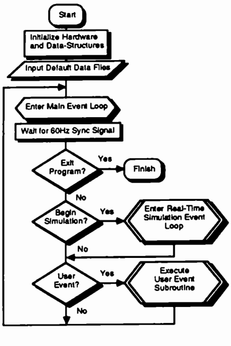

Start
Initialize Hardware and Data-Structures
Input Default Data Files
Enter Main Event Loop
Wait for 60Hz Sync Signal
Exit Program?
Yes
Finish
No
Begin Simulation?
Yes
Enter Real-Time Simulation Event Loop
No
User Event?
Yes
Execute User Event Subroutine
No

Fig. 4 Main Event Loop

Real-Time Simulation Event Loop

The primary functions of the EOS (such as command, control, and monitoring) must be operational at the same time the simulator is being flown if they are to be of any use to an experimenter. Since the simulator operates at a frequency of 30 hertz, the EOS must be able to record data at that rate while simultaneously responding to user requests as well. The Real-Time Simulation Event Loop is designed to operate under the difficult conditions imposed by an interactive, real-time environment. Entry into the loop occurs whenever a simulated flight begins and exit from the loop occurs whenever the simulated flight ends. While in this loop, all procedures which require more execution time than the allowable thirtieth of a second are rendered inaccessible. Unlike the Main Event Loop, which returns to the point of loop departure immediately following execution of each procedure (Fig. 4), the Real-Time Simulation Event Loop uses an event priority algorithm to determine which functions are called during any given pass through the loop and returning to the top of the loop following execution of those procedures. This method (illustrated in Fig. 5) ensures that total frame time does not exceed one thirtieth of a second. A timing diagram (Fig. 6) is used to demonstrate why this kind of priority algorithm is needed. In order to use such a technique, all possible events must first be identified and assigned a priority.

The highest priority event (which is also the most time critical function of the EOS) is recording simulator output parameters at the same rate they are being generated (30 hertz). This recorded information forms raw data files which are necessary for post-flight statistical analysis. Since the simulation frame time is one-thirtieth of a second and the IRIS' screen refresh interval (marked by a sync signal) is one sixtieth of a second, a repetitive procedure which reads a frame of data from the Dual Port Memory must be executed once after every other sync signal. If this were all that the EOS were required to do, then its tasks would be extremely simple to program. Unfortunately, this is not the case. There are multiple tasks which must share the CPU resources such as user-requested events, and periodic updating of on-screen plot displays. Although data recording events are always given the highest priority, some method for handling these additional procedures without disturbing the critical data recording cycle is required.

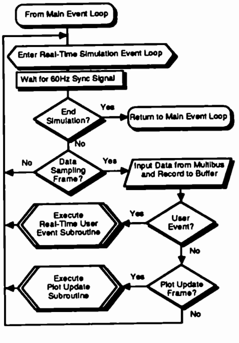

From Main Event Loop
Enter Real-Time Simulation Event Loop
Wait for 60Hz Sync Signal
End Simulation?
Yes
Return to Main Event Loop
No
No
Data Sampling Frame?
Yes
Input Data from Multibus and Record to Buffer
Execute Real-Time User Event Subroutine
Yes
User Event?
No
Execute Plot Update Subroutine
Yes
Plot Update Frame?
No

Fig. 5 Real-Time Simulation Event Loop

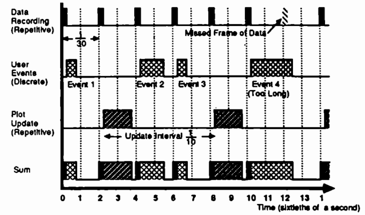


Fig. 6 Procedure Execution Timing Diagram

The type of events which have the second-highest priority are the user-requested events. Most real-time user-requested events are simply flag-setting Command and Control functions which are relatively short procedures. There are, however, other user events (such as selecting a new plot for display or bringing another set of menu items up on the screen for selection) that require nearly the full thirtieth of a second to process. In order to give each user event the maximum amount of time for execution, these events are only processed after a data recording event. Following the completion of the event, the flow of control immediately returns to the top of the event loop and awaits the next sync signal. As long as the user event is completed prior to the next data-recording sync signal, the recording cycle will not be disturbed.

Events of the lowest priority are on-screen plot update events since the screen resolution is so low that the plots need not be updated any faster than about ten times every second. If there is not a pending user event immediately following a data record event, the program checks for a pending plot update event. If a plot update event is detected, then it will be executed, otherwise the program again returns to the top of the loop. Thus, a plot update event may only occur immediately after a data recording event and only if a pending user event is not present. At first it may seem that very little plotting of data will be done using this scheme, however, user events are relatively rare and therefore missing an update even at the rate of once every second will have virtually no visible effect on the display.

Occasionally a user event procedure will require more time than is allowable between data recording events. For example, if a user event procedure is not completed in time, then a diagnostic warning is issued identifying that particular procedure as taking too long to process. If its execution time cannot be shortened by reprogramming the procedure, then it may be possible to break it down and execute it in increments over a sequence of successive loop passes. This latter method involves use of an event queue which is a collection of event flags representing procedures to be executed. Each pass through the loop, the program enters the procedure identified by the first event flag in the queue and executes a small segment of that procedure. When the procedure has been fully executed, its flag is removed from the queue and the procedure represented by the next flag in line is executed, segment by segment. When all segments of all procedures listed in the event queue have been completed, the program accepts new user events again.

User Interface

All communication between the EOS and human operators is handled by a set of routines which collectively make up the User Interface. These routines are designed to:

1. inform the user about available options,

2. allow the user to indicate his selection to the EOS,

3. provide immediate feedback, confirming which selection has been made, and

4. display the information associated with the user's request (i.e., data plots, maps, help information, etc.).

Screen Display

The most noticeable aspect of the User Interface is the screen display layout which is pictured in Fig. 7. In order to provide the most effective use of screen real estate the EOS divides the 1024 x 768 pixel area into six major subareas each of which is responsible for a different input/output function.

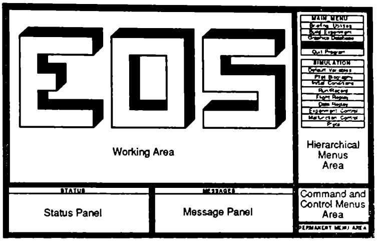


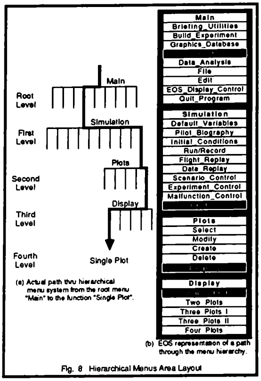
Fig. 7 EOS Screen Display Regions

Working Area

The largest area is a region which is used to display user-requested information such as data plots, tactical map, repeater instruments, and help information. It may also be used to display special types of menus associated with creating plots, selecting tactical map options, or choosing additional items from a help index. For greater flexibility in simultaneously displaying multiple data plots of various sizes, the Working Area may be divided into one, two, three, or four windows. This gives the operator a choice between displaying many low resolution data plots or a few high resolution data plots.

Hierarchical Menus Area

The Hierarchical Menus Area is used to display various levels of menus in the menu hierarchy and the path taken to reach a given menu. Most of the menus in this area are presented in the standard vertical format. The Main Menu (visible at all times) is the root of the hierarchy and consequently occupies the top position in this window. As each new menu is entered (by descending down the hierarchical path), it is displayed below the preceding menu. Fig. 8 illustrates this concept and shows how the display in the Hierarchical Menus Area correlates with the path of selections made by progressing through the hierarchy to a given menu item or function.



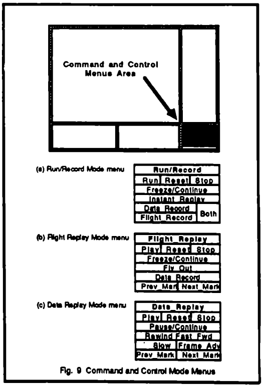
(a) Actual path thru hierarchical menu system from the root menu "Main" to the function "Single Plot".

(b) EOS representation of a path through the menu hierarchy.

Fig. 8 Hierarchical Menus Area Layout

Command and Control Menus Area

Since the necessity for maintaining complete control of the simulator is of the highest priority, the many Command and Control functions are selected from menus which are located in a separate, dedicated area directly below the Hierarchical Menus Area. There is a different menu for each of the three Command and Control modes: Run/Record, Flight Replay, and Data Replay (Fig 9). Only one of these menus may occupy the Command and Control Menus Area at any given time and once displayed, cannot be removed



(a) Run/Record Mode menu

(b) Flight Replay Mode menu

(c) Data Replay Mode menu

Fig. 9 Command and Control Mode Menus

while the simulation is actively being flown.

Permanent Menu Area

This area contains menu items that the user requires quick access to at all times during EOS operation (such as Help, Display Paging, etc.) and hence they are permanently located in the bottom right portion of the screen.

Message Panel

This area is a standard UNIX input and output textport which allows the user to input standard alphanumeric text strings and numbers from the keyboard and allows the program to write various messages back. During development, this textport is extremely useful for debugging purposes.

Status Panel

This panel is used to provide useful information about the status of the EOS system. It contains current time and date, the names of the pilot, instructor, and experiment being run, total elapsed flight time, total elapsed session time, current record or playback file name, current frame of data being recorded or replayed, and a bar graph showing the size of the record or playback file.

Conclusions And Recommendations

An Experimenter Operator Station which provides Command and Control, Monitoring, and Utility capabilities has been designed and implemented in the C language on an IRIS 2400 Turbo workstation. Structured programming design methods were used to develop software that is efficient, modular, and designed for future expansion. A user interface with a hierarchical menu system has been created which makes effective use of the screen's viewing area and provides mouse-oriented, menu-driven control for virtually every function. All real-time functions have been shown to operate interactively at a frequency of 30 hertz for the UH-1TRS simulator configuration or 15 hertz for the UAFDL configuration. The EOS was designed using existing simulator hardware only and is compatible with both the UH-1TRS and UAFDL simulator configurations.

Future Options

Many potential features were considered in addition to those actually included in the initial version of the EOS. Room for many of these has been designed into the EOS software framework. A few of the more obvious enhancements in the area of monitoring functions would be the addition of a Tactical Map and a display of Repeater Instruments. The concept of a Tactical Map display is a variation of the standard Latitude vs. Longitude or Groundtrack plot. It would show a projection of the aircraft on the underlying terrain. Unlike a simple groundtrack plot which only provides an indication of the two dimensional position of the aircraft, the Tactical Map display could use graphical symbology (much like an air traffic controller's screen) to represent the three-dimensional position, attitude, velocity, and rate of climb of all vehicles in the simulation. In addition to these multiple vehicle indicators, the map could also display (at the discretion of the operator):

1. wind direction and turbulence conditions,
2. IFR approach plates,
3. electronic beacons and other navigational aids,
4. physical landmarks,
5. ideal flight path (for traffic patterns or designated IFR routes), and
6. distance from designated way points.

A most helpful feature of such a map would be a zoom or magnification capability in order to obtain higher resolution during slow speed flights or hover and taxi maneuvers. The Tactical Map could also be used to set initial conditions using the mouse and cursor rather than typing in explicit values from the keyboard. For instance, to set the wind direction, one would simply move the wind direction arrow to the desired heading with the cursor, or to set the initial latitude and longitude, one would simply move the cursor to the desired location on the map and press a mouse button.

The Repeater Instruments display would be used to provide a color graphic representation of the pilot's instrument cluster which would allow an instructor pilot or operator to monitor the feedback from the same set of instruments which the pilot in the cockpit is using. In addition to the Tactical Map and Repeater instruments, other potential EOS features and enhancements which have not yet been implemented into the EOS include:

1. an on-line Help Facility,
2. transmission of more flight variables to the EOS and selectable control over which variables will be recorded,
3. real-time control over the graphical database attributes such as moving target vehicles, weather effects, and time of day,
4. extension of the User Interface to provide keyboard equivalents for mouse-operated selections,
5. integrated control of data file management and statistical post-processing through the EOS User Interface,
6. pilot briefing and debriefing utilities, and
7. control of simulated malfunctions.

Additional possibilities for the EOS are expected to be discovered as researchers become more experienced with its present capabilities and developers begin to enhance the system with some of the features presented here. It is unquestionable, however, that the initial version of the Experimenter Operator Station is a powerful addition to the UH-1TRS system and will greatly enhance the researcher's ability to study the effectiveness of simulator pilot training.

REFERENCES

1. Rolfe, J. M. and K. J. Staples, Flight Simulation, Cambridge: Cambridge University Press, 1986.

2. Rhyne, R. E., Identification of Helicopter Simulator Dynamics, Tuscaloosa, Alabama, 1987.

3. Bailey, J. E.; K. Krishnakumar; J. E. Ward, IV; and T. A. Kaye, UH-1 Training Research Simulator Automated Pilot Measurement and Graphics Data Base Visual Cue Assessment, BER Report No. 400-177, September 1986.

4. Dudgeon, J. E.; J. J. Dearth; L. M. Freeman; J. E. Bailey; T. F. Gatts; and H. Sunsarsa, VICOPS Helicopter Simulator Architecture and Hardware Design, BER Report No. 376-29, March 1986.

Complete source code listings for the Experimenter Operator Station are fully documented in the Bureau of Engineering Research publication entitled Design and Development of an Experimenter Operator Station for the UH-1TRS Simulator, BER Report No. 376-29, November 1988. This report may be obtained from:

The University of Alabama
College of Engineering
Bureau of Engineering Research
Box 870200
Tuscaloosa, AL 35487-0200
Telephone: (205) 348-1591

A VOICE-OPERATED INSTRUCTOR'S STATION

Richard Fox and Tony Weaver
Link Flight Simulation Division of CAE-Link Corporation
Binghamton, N.Y.

ABSTRACT

Whether a mission is one involving subsonic or supersonic aircraft, the relative proximity of the combatants requires that aircrew members react to events in a fraction of a second and therefore that the instructor must also display similar reaction times. A more natural control mechanism must be developed to support interaction of the instructor with the simulator and to reduce the complexity of his interactions with the simulator controls (usually an instructor/operator console), reducing the directly related induced stress. This would allow the instructor to concentrate on the training of the student rather than the manipulation of a complex simulation system. One proposed solution is to employ a voice recognition system which would allow the instructor to communicate with the simulator controls via simple voice commands, thus eliminating the physical gymnastics currently required. The instructor would have the capability of updating trainer data, states, modes, etc. with a minimum of voice commands.

INTRODUCTION

Pilot training today is looked upon not as a luxury, but as a necessity. Poorly trained pilots are a danger to themselves, a danger to their crew, and a danger to our national security. In order to increase the survival rate of the aircraft and the pilot, careful actions and reactions must be ingrained into the pilot. Thus, flight simulators have become more of a necessity. Obviously, it is much better for a pilot to crash or be shot down in a simulator than in a multimillion-dollar aircraft. In order to duplicate combat missions, simulators have to be sophisticated. Unfortunately, in an attempt to become more realistic they have become complex to the extent that one person can not assimilate the real-time information fast enough to react, and as a result, the complexity of simulation has had an overloading effect on the instructors. The number of alternative decisions available to the pilot at a given time is extensive; however, this is surpassed by the number of decisions that the instructor has available. Thus, our goal is to increase the quality of training for the pilot while simultaneously increasing the efficiency of the training device. One means to achieve this is to aid the instructor, who is at this point irreplaceable.

The instructor has two obligations: 1) to observe, and at some point, correct the student, and 2) to engage the student in real-life situations. The instructor should not be overburdened by simulator operations. When he is operating the simulator he is probably not "instructing." What is needed is a "natural" and efficient interface between the instructor and the instructor/operator station (IOS).

Speech is a natural way of communicating our needs. Speech recognition technology is well suited to the trainer environment since both are real-time. Also, speech recognition is capable of tolerating data which may be erroneous to the IOS system, which is common in the trainer environment. However, speech recognition is unique when compared to the trainer, in that the speech systems are capable of self-adaptation.

Voice has always been a technical challenge, and would continue to be in a simulation environment. We should not expect complete speaker independence and unrestricted universal speech; however, limited speech recognition exists today and would be useful in many areas of simulation. The area which is most appealing is the IOS. To stay consistent with our primary goal of natural communication, we wish to build a speech system which has been trained to respond to instructor needs, not one which requires extensive instructor training.

IOS OPERATIONS

In the past the IOS was a collection of mechanized dials, mechanical readout devices, and cockpit repeaters. These instruments are slowly being replaced by CRT pages, and it is not uncommon to have 300 to 600 pages for a particular simulator. The instructors are required to understand how to use each of these pages. However, there is an attempt to logically order the pages. As an example, aircraft armament information is stored on a Stores Loading Page, while target information is stored on a Target Manual Page. Also, great effort is taken to limit the number of keystrokes required to move from one page to another. Unfortunately, with all this, the instructors are still placed under tremendous pressure to recall the details of the IOS when their concentration should be on the student pilot.

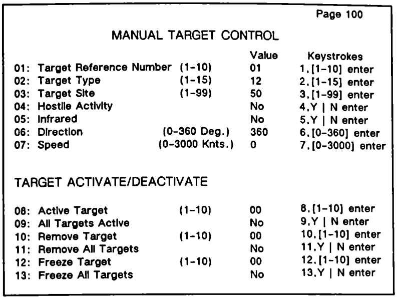
Man-machine interfaces have been implemented in the last ten years to facilitate keyboard-instructor interchange; these include the mouse and the light pen. However, even though both of those input devices are more sophisticated than the keyboard, they still require eye-hand coordination, and neither of them aids the instructor in handling the IOS.

A generic IOS page is shown in Figure 1.* The thirteen command entry lines are used to select, activate, and control the target's characteristics. These characteristics are usually programmed prior to the beginning of the mission, but at any time may be reprogrammed by the instructor. Each command is entered on the keyboard according to the Keystrokes column on the page. There is a minimum of four keystrokes to complete an entry and many of the commands must be entered as a unit. For instance, a change in target type may necessitate a change in target speed since different targets have different ranges of speed.

A voice system would operate differently than the present-day IOS. First, when a page is displayed the voice system sets its "page context." Page context is knowledge information about the pages. Not only could the instructor alter lines on a page by the present-day keystroke method,

* For a complete description of the Manual Target Control page see Appendix B.

he could also alter them by a variety of voice commands. For instance, under the Target Activate section the Target Reference Number (Figure 1) could be accessed by stating "Set Target Reference Number to 2" or "Set number 3 to 2." These commands are synonymous because the Manual Target Control page is currently being displayed (the page context is set to Manual Target Control). If the page is not currently displayed, the system would have an incorrect page context and an error would occur. The instructor could switch the page by stating "Display page 100," or "Display Target Manual Control page." If the instructor was really familiar with the pages, he could state "Page 100, change Target Reference Number to 3." The commands and the updated page would be displayed.



```
                                                Page 100
                  MANUAL TARGET CONTROL
                                         Value    Keystrokes
01: Target Reference Number (1-10)       01       1,[1-10] enter
02: Target Type             (1-15)       12       2,[1-15] enter
03: Target Site             (1-99)       50       3,[1-99] enter
04: Hostile Activity                     No       4,Y | N enter
05: Infrared                             No       5,Y | N enter
06: Direction          (0-360 Deg.)      360      6,[0-360] enter
07: Speed             (0-3000 Knts.)     0        7,[0-3000] enter

TARGET ACTIVATE/DEACTIVATE

08: Active Target           (1-10)       00       8,[1-10] enter
09: All Targets Active                   No       9,Y | N enter
10: Remove Target           (1-10)       00       10,[1-10] enter
11: Remove All Targets                   No       11,Y | N enter
12: Freeze Target           (1-10)       00       12,[1-10] enter
13: Freeze All Targets                   No       13,Y | N enter
```

Figure 1 Typical IOS CRT Page

Our first step would be to determine which pages would yield the greatest payback in terms of making the system more natural. Obviously, some situations place more stress on the instructor than others. It is much easier for an instructor to reason while setting up the trainer for a mission, rather than in a time-critical situation, such as an air-to-air conflict. Therefore, the first tests would be in a real-time situation using pages from the Stores Management System, the Fire Control Computer, Flight Control Computer (for the targets), Emitter Stations, CCM, Malfunction, and Environmental Conditions.

SPEECH SYSTEMS

In recent years true voice recognition has made significant progress. Machines can presently translate similar phrases, punctuate them correctly, and perform required tasks. Some systems use context-sensitivity to understand continuous speech. Other systems claim to understand vocabularies up to 15,000 words.[6]

A Voice Recognition system first translates the voice features into data and then tries to match it to a pre-recorded example (sometimes called a "voice template"). Different systems employ different techniques to match the voice template. Humans use many approaches, such as context sensitivity, statistical averaging, phonetic matching, and human anticipation. The best system should use all of these. From the standpoint of this paper it really doesn't matter which method is used for matching (although some are more plausible than others), only that the chosen system coincides to the requirements listed later. The characteristics of voice systems which are most visible are whether they are speaker-dependent or speaker-independent, and whether they are isolated or continuous speech recognizers.

A speaker-dependent system is one that can understand only one person's voice template. A speaker-independent system will recognize more than one voice template. Speaker-dependent systems have a further restriction in that they require training. It would be desirable to use only speaker-independent systems; however, these systems have a restricted vocabulary, sometimes as small as the digits zero through ten. It might be necessary to have a hybrid of the two technologies, a reasonable size vocabulary with certain critical words being independently recognizable.

Some systems recognize isolated words (discrete utterances), others recognize connected words (continuous speech), and some do both. For isolated words, voice systems have higher recognition performance than humans. Early systems were dependent on silence separating the words, a problem since some people slur one word into the next word. An extremely fast talker would prove to be a problem. Realistically, some speakers are going to have to be semicontrolled. In this regard, our major objective is to place as little burden as possible on the instructor.

Continuous speech is difficult because the system must listen for clues of context. Distinguishing between phrases like "Today's Monday" or "hurray its fun day" and "Speech recognizer" or "beach is nicer," as someone randomly speaks them over a telephone is not easy for man or machine. The key to continuous speech is context sensitivity. Context sensitivity confines the search space for patterns so that some groups of words have a higher probability of occurring than others. This idea is analogous to a parse tree, where some words have a higher probability of being spoken in different branches of the tree. Some voice systems take this further and define a grammar that can accept or reject utterances (or group of utterances) based on a given situation. Any grammar is, of course, environment-dependent, and the grammar for an IOS instructor is different from that for a radiologist.

ADAPTIVE SYSTEMS

Voice recognition, which is a particular case of pattern recognition, poses an immense programming effort if conventional methods are used; fortunately, there are alternative methods. The one examined in this article classifies the audible commands using adaptive algorithms. These algorithms are based on programming by example. The advantages of adaptive systems over conventional systems are:

- High fidelity in pattern matching can be obtained without real programming being performed on the pattern to be classified.
- The recognition algorithm is very efficient in the classification of the pattern.
- With the correct training set, a high number of correctly classified patterns are possible.

To explain the adaptive system method briefly,* we simply classify a pattern into a particular class partition. A pattern may be defined as the sum of its attributes and features. A particular pattern falls somewhere in n-dimensional space. The purpose of training the system is to force the

* For a detailed description see Reference 1 (Nilsson).

patterns to fall in a specific space, in our case the space that coincides with the voice template (see Figure 2 for adaptive system in 3-D space).

For any technology there are always disadvantages. The major disadvantage for adaptive systems is in making the system robust enough to allow for small changes in its patterns vs. changes in its environment. It would be a disadvantage if the instructors were constantly training the machine simply because the machine cannot handle small changes in its environment.

Relating adaptive systems to the voice recognition system is simply a matter of substituting the voice template into the classical adaptive system development methodology. The basic steps in this methodology are as follows:

- Development of the subject and its categories. In our case it is utterances from instructors to the IOS.
- Select the features that are most likely to be important. A subset for voice recognition is pitch, accent, amplitude, etc. These features are characteristics of the voice system chosen.
- Train the machine to recognize the utterances.
- Use the system in actual application or run the machine on another training set to determine the performance of the machine.
- Reiterate the above steps until an acceptable performance level is reached.

AN IOS VOICE RECOGNITION SYSTEM

Figure 3 is a diagram of a representative voice system and depicts the basic flow of information within the training environment. The voice recognizer will contain the voice templates for the speaker. The instructor will speak into a voice recognizer which will translate the pattern into a predefined word code, which is simply a number. The situation analyzer will monitor the state of the trainer and relate any information that will facilitate choosing grammar branches (the information might be that missile number one is ready to fire). Any keystrokes are combined with the above information in the command synthesizer, which will either accept the command and issue IOS commands or reject the command and possibly ask for more information from the instructor. The voice synthesizer holds the parse trees. The performance monitoring station will record the voice system behavior.

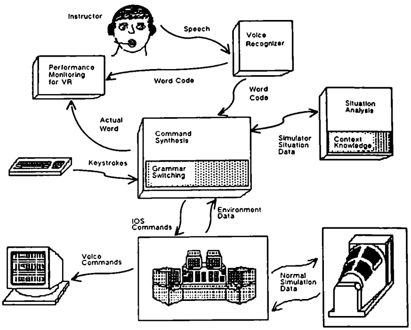


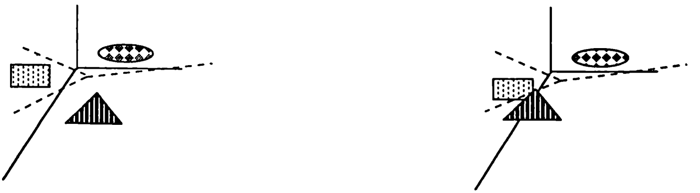
Figure 3 An IOS Voice Recognition System

IOS VOICE SYSTEM DESIGN CRITERIA

Five basic criteria must be met in order to complete the primary objective of a natural IOS interface. All the criteria are interrelated, and failure to meet any one of them will severly impact the others' effectiveness.

First, the voice recognition system must be much easier to learn to use than current systems such as touch-screen or mouse systems. It should be intuitively obvious to operate, communicate with, and control. It should take little training to learn and minimal effort for recalling and issuing IOS commands. If the system fails to meet this foundation requirement we would only be trading new problems for old while solving nothing.

Second, since instructors are in high demand, any extra demands on their time should be kept to a minimum. It would not be desirable for a instructor to take many hours for the sole purpose of training the machine. One solution is to have them speak their commands as they enter them at the keyboard in the present-day system. This scenario will yield real-life voice templates with minimum effort. To take this a step further, when the system begins to perform poorly, the voice system can automatically re-record the problem voices and assimilate them into the old voice template.



The diagram above depicts the possible dichotomies between three classes of patterns. If we assign the square all utterances that sound like the word "one," the ellipse "two," and the triangle "three," a pattern when identified will be placed in one of these classes, hopefully the correct one. If not correctly placed the machine will be retrained for those problem words.

The above dichotomies are not completely separated because two of the three classes overlap. This will yield incorrectly identified patterns. One means to correct this is to alter the feature selections to be more distinguishing.

Figure 2 3 Pattern Classes in 3-D Space

The third criterion deals with the critical steps in the voice system development of the grammar and system command syntax, which should be as brief as possible, as long as the commands are not in conflict with making the system natural for the instructor. Where acronyms are commonly used, the system should be able to handle the full name and the acronym, such as "display SM" or "display Stores Management page."

Instructors should be the main influence on the construction of the grammar and system commands syntax; however, the syntax will go through many alterations during its lifetime, since it is important to realize that during pressure situations people tend to shorten explanations with slang such as "Kill three" for "Fire missile 3."

In many conversations it is not necessary to recognize every word in a sentence. For this reason there are four, possibly interrelated, sets of words that would be essential for recognition (see below). As an example consider the command "Move target 10 to Lat. 10, Long. 5."

1) Context Distinguishing Words - (Lat, Long, target)

2) IOS Commands Words - (Move) others include Fire, Arm, Load

3) Attribute Words - all the numericals (one, ten, hundred,...) and other included variables.

4) Voice System Command Words - such as Undo, Execute, Run, Erase, etc.

The fourth criterion is that a voice-independent system is more desirable than a dependent one; however, it is apparent that technology is not yet advanced enough to present such a system, so we would compensate for an "almost voice-independent" system. This means that the system must understand a "voice switch," in other words, two or more people speaking conversationally. In a training environment a reasonable limit for the number of speakers is five. Let's say that at a training base there are 200 different voice templates (200 instructors) that the machine must recognize; however, out of those 200 only five may be active a(any one time. The system would work by simply downloading the five voice templates of the training crew assigned for that mission.

To accomplish the voice switch, some voice systems combine voices into an "average template." We do not believe this is effective in this instance, since you are never sure of the personnel of a particular instructor training set for any given day. Out of the 200 instructors you do not know which ones will be working together and, of course, you do not want to average every permutation. It seems to be much more efficient to download each voice template and analyze each one independently. The crew would download their voice patterns with their normal start-up procedures.

The fifth and final basic criterion is that we would, of course, require the ability to recognize isolated words and continuous speech. To take this a step further, we would like the system to be capable of understanding spoken words prior to completion of the sentence command in order for the instructor to verify as soon as possible that the recognized command was correct. If the sentence has a mistake before completion of the command, the instructor can correct it at that point or issue the voice system "erase command" and start over.

An observation with voice is that people tend to demand real-time response. When you speak to someone you have a "waiting window" in which you expect acknowledgement. Waiting windows are totally dependent on the listeners. It is reasonable to expect the system to repond in less than one-half second in recognizing the command. This time limit refers to the time it takes to recognize and process the command and to transfer it to the simulator flight systems. At first glance one-half second might seem slow; however, given the fact that current IOS systems require many selections, performed by hand movements, the half-second requirement to perform all its processing is acceptable when viewed within the time frame of the total selection process.

At this point we can relate the above criteria to the voice system's functional development. The major functional phases are Voice Recognition Development, Template Development, and Voice Recognition Interaction. The Voice Recognition Development phase identifies overall system characteristics, the Template Development phase describes utterances and grammar construction criteria, and the Voice Recognition Interaction phase is concerned with the interchange between the instructor and the voice system. The requirements applicable to each of these phases are as follows:

1) VOICE RECOGNITION DEVELOPMENT

a) The system should have separate processors for the speech recognition process. Each processor will have the ability to store a voice template.

b) The trainer should perform exactly the same whether it is commanded from a speech processor or the traditional keyboard.

c) The system must possess an overall command recognition performance of 98%3, with 100% correctly recognized commands on the second try. Furthermore, the system must have a performance above 98% for first-time recognition of crucial words.

d) The system vocabulary size has yet to be determined; however, it is estimated that the requirement would not exceed 2,500. words.

e) For display purposes not all utterances appear. Utterances that have no command discriminating features, and thus are non-essential, should be disregarded.

f) The system should contain some form of context sensitivity. This could be in the form of voice menus, grammar branches, and parse trees.

2) TEMPLATE DOCUMENT

a) A majority of the training of the voice recognizer will be from the recordings of the instructors' voices during an actual flight crew training mission.

b) During evolution of the system the grammar will change; therefore, a minor change in vocabulary should be performed without retraining for other non-related commands. This would also be the case if one particular word becomes troublesome.

c) There should be a conflict matrix indicating troublesome utterances. The problem utterances can be dealt with by either changing the vocabulary or the grammar. It should be noted that the contents of the conflict matrix must be interpreted carefully in designing system changes. Problems can be either user-dependent or user-independent.

Changing the grammar for one instructor may cause problems for another instructor.

d) During training it would be desirable to have as much control over the recognition process as possible. Two criteria hold: 1) observation of the grammar using graphical representation of the syntax of the language, and 2) being able to massage the voice template in accordance with the conflict matrix.

3) VOICE RECOGNITION INTERACTION

a) Of course, we would like to have the voice system ignore the instructor's conversational words. This could be accomplished by either physically turning the system on and off (by voice command or manual switch) or, ideally, this would occur naturally as a side effect of the grammar.

b) As a visual check, the interpreted voice command will be displayed on a CRT. The words should be displayed as they are spoken, rather than an entire command being displayed after acceptance.

c) There will be a confirmation of critical commands.

d) The instructor should have the ability to program macros or aliases for sets of words. This will allow him to tailor the grammar to his liking while shortening commands he feels are too long.

e) The system should have the ability to capture all or part of the dialogue from the instructor to the IOS and place it in a history file. A transaction processor can roll back and alter or replay any of the commands. The instructor could also re-issue any command from this history list. The history file would be saved on disk for replay at a later date.*

f) There should be an on-line help system. The help system can be somewhat tailored by the instructor. The system will allow an instructor to choose between a terse error explanation or an expanded error explanation. In other words, if a syntax error occurred during a command, the system would display the command and the error in bold print followed by either a terse explanation, such as "syntax error", or an expanded version giving a more detailed explanation and maybe even alternatives to that command. If the system has only one path down the parse tree, the instructor might allow the voice system to complete certain commands. This will enable automatic minor error correcting, such as "load gun one" being substituted for "load gun gun." However, there are times when the system cannot correct the error and human intervention is required. At such times, the system will issue the error and display the questionable phrase highlighted in bold, such as

"Syntax Error: **hold** gun one."

g) The system should have an automatic monitoring system to detect when low performance is occurring and relate this at the end of a mission.

ISSUES

One danger area is in the development of the grammar. The language defined could be natural to one instructor, but alien to another. To minimize the dilemma, the command set must always be driven to contain generic universal commands. It will never be perfect, but ideally, the set of aliases that would be created by each instructor could compensate for minor idiosyncratic changes in the syntax.

Another potential problem is that the system may not be all-encompassing, meaning that there are always technology problems that inhibit fulfillment of an objective. The system may not handle all the necessary voice commands and therefore fall short of the major objective of creating a natural environment. So, some commands might be forced to be entered via the keyboard, when in a perfect world they would be voice commands.

There is one performance issue that should be addressed. That is the problem of making the voice system robust (being able to recognize commands when the instructor has a common cold). However, an extremely robust system would cause another problem with false alarms that might trigger major simulator events — Such commands as "fire one" when the instructor actually said "fire gun." This problem could disrupt or even scrub the entire training mission. Every effort must be put forward to prevent this situation. This implies that not only must the system have a high performance in accepting words, but also a high rejection accuracy if the word does not fit the context. The potential for this problem is reduced with the presence of the grammar switching and situation analysis systems combined with confirmation of critical commands.

One of the major dangers during training of the voice recognizer is creating a system in which the instructors become trained rather than the machine. The instructor might unconsciously alter his voice and create a false high performance in recognizing his voice. In order to avoid this situation the training should come from a real-life training exercise. This might best be accomplished by a mock mission in order to train the machine. However, this would be in direct conflict with one of our requirements that most of the training should be offline from the simulator. Thus, composing a "true to life" training set for the instructor voice is a major concern.

CONCLUSION

A voice recognition system appears to be applicable to the problem of making the IOS more natural, and thus increasing the quality of training for the pilot. It appears that the technology is available and at a level to meet most of our goals. The next step is to build a prototype to verify the capabilities and discover the limitations of such a system.

* See Appendix A for voice recognizable system commands.

APPENDIX A

Voice Recognizable commands:

1) delete [word] [last #action]
2) execute [#command]
3) alter [#command]
4) display [#command]
5) change [old command] new command
6) stop #command
7) continue #command
8) store #command as variable-name
9) alias #name #name

APPENDIX B

Target Manual Page Description

01	Target Reference Number	This is used to count the number of targets being displayed.
02	Target Type	The Target Type is code which identifies the characteristics of the emitter. This would be displayed on another page of the IOS.
03	Target Site	The Target Type identifies a predefined target location.
04	Hostile Activity	Make the target capable of attacking.
05	Infrared	Identifies whether the target is hot or cold.
06	Direction	Indicates in which direction the target is traveling with respect to true north.
07	Speed	Indicates the speed at which the target is traveling (zero is valid).
08	Active Target	Activates or deactivates the target displayed in item 1.
09	All Targets Active	Activates or deactivates all targets.
10	Remove Target	Delete the target displayed in item 1.
11	Remove All Targets	Delete all the targets.
12	Freeze Target	Freeze the target displayed in item 1.
13	Freeze All Targets	Freeze all targets.

REFERENCES

1. Nilsson, Nils, Learning Machines. McGraw-Hill Inc., 1965
2. Reddy, Raj, "Foundations and Grand Challenges of Artificial Intelligence." AAAI (Winter, 1988): 9-20.
3. Rosch, Winn L., "Understanding the Master's Voice," PC Magazine (Oct. 1987): 261-308.
4. Lectures by Donald Gause; Systems Science Dept., SUNY at Binghamton, 1985.
5. Simpson, Carol A., Bunell, John W., Krones, Robert, "Smart Command Recognizer (SCR): For Development, Test, and Implementation of Speech Commands." American Institute of Aeronautics and Astronautics Inc. (1988): 215-221.
6. Information brochures from Kurzweil AI Inc., 1988-89.
7. Capt. Williamson, David, Small, Ron, Feitshans, Gregory, "Speech Technology in the Flight Dynamics Laboratory," Air Force Wright Aeronautical Laboratories (AFWAL/FIGR).
8. Pallett, David, "Performance Assessment of Automatic Speech Recognizers," Journal of Research of the National Bureau of Standards, Volume 90, Number 5, (September - October, 1985): 371-387.
9. Simpson, Carol A., Navarro, Teresa, "Intelligibility of Computer Generated Speech as a Function of Multiple Factors." IEEE (1984): 932-940.
10. Malkin, Frank, Dennison, Thomas, "The Effect of Helicopter Vibration on the Accuracy of a Voice Recognition System." IEEE (1986): 813-817.
11. Lowen, Walter, Dichotomies of the Mind. John Wiley & Sons, Inc. 1982.
12. Rich, Elaine, Artifical Intelligence. McGraw-Hill, Inc. 1983.
13. Simpson, Carol A., Dr. Ruth, John C., "Phonetic Discrimination (PD-100) Test For Robust Speech Recognition System." IEEE (1987): 889-896.
14. Roberts, Darel, Roberts, Jefery, "VIOLET — A Voice Input/Output Lexically Endowed Terminal as an Aid to the Physically Handicapped Computer User," Roberts Enterprises, Sepulveda, CA, no date.
15. House, Arthur S., Williams, Carl E., Hecker, Michael, Kryter, K. D., "Articulation-Testing Methods: Consonantal Differentiation with a Closed-Response Set." The Journal of the Acoustical Society of America, Volume 37, Number 1 (January 1965): 158-166.

ABOUT THE AUTHORS

Richard Fox is a System Software Engineer working in the Computer Systems Department at the Link Flight Simulation Division of CAE-Link Corporation in Binghamton, New York. He has received his M.S. degree in System Science from the State University of New York at Binghamton, and a B.S. degree in Computer Science from Rochester Institute of Technology in Rochester, New York. He has spent 6 years at Link researching new technologies with applications to the training environment.

Tony Weaver is a staff scientist with CAE-Link Corporation and is currently assigned to the IR&D group concentrating on Micro Simulation Technology (MST). He has 25 years experience at Link in both technical and management positions. Prior to MST he supervised an IR&D group dealing with the issue of modern software engineering (2167A, Object Oriented Design and Ada). Tony attended Broome Community College and Alfred University and has concentrated his efforts in the field of computer science as applied to flight simulation, including networking, distributed processing, and artificial intelligence.

89-3306-CP

AN IMPROVED NUMERICAL INTEGRATION METHOD FOR FLIGHT SIMULATION

R.M. Howe*

The University of Michigan
Ann Arbor, Michigan

Applied Dynamics International
Ann Arbor, Michigan

Abstract

In this paper a modified form of Euler integration is described which, when applied to the six-degree of freedom flight equations, retains and enhances many of the advantages of AB-2 integration and at the same time eliminates the disadvantages. The scheme is based on the Euler integration formula, but with the state-variable derivative represented at the midpoint of each integration step. In this case the conventional first-order Euler method actually becomes second order, with a very small accompanying error coefficient. To apply this method to the six-degree-of-freedom flight equations it is necessary to define velocity states at half-integer frame times and position states at integer frame times. It is shown through dynamic error analysis that the modified Euler method has an error coefficient which is one-tenth that associated with AB-2. The method also exhibits minimal output delay in response to transient inputs. The modified Euler method may also be useful in the integration of state and costate equations in real-time mechanization of Kalman filters for navigation and control systems.

1.Introduction

The ever increasing complexity of the math models used as a basis for real time flight simulation has continued to apply pressure on digital processor speed requirements for such simulations. More effective numerical integration algorithms can help relieve some of this pressure. The most popular method currently in use for flight simulation is the Adams-Bashforth second-order predictor method, usually referred to as AB-2. Its advantages include second-order accuracy with respect to integration step size, only one required pass through the state equations per integration step, and compatibility with real-time inputs. Disadvantages include stability problems associated with extraneous roots and response delays of one or two frames following transient inputs.

In the next section we consider AB-2 along with two other second-order integration methods suitable for real-time simulation of dynamic systems. The first is a two-pass real-time predictor-corrector algorithm and the second is a single-pass version of the same. We then introduce the modified-Euler method and describe its application to the flight equations. The accuracy of the various methods is compared by means of time-history plots of the dynamic error in simulating aircraft response to a control-surface input.

2. Some Real-time Second-order Integration Algorithms

Consider first the AB-2 predictor integration method applied to the state equation given by

$$\dot{X} = F[X, U(t)] \qquad (1)$$

Here X is the state vector, U is the input vector, and h is the integration step size. The standard AB-2 predictor algorithm is the following:

$$X_{n+1} = X_n + h\,(1.5\,F_n - .5\,F_{n-1}) \qquad (2)$$

where $X_n = X(nh)$ and

$$F_n = F(X_n, U_n)\,, \quad F_{n-1} = F(X_{n-1}, U_{n-1}) \qquad (3)$$

The AB-2 formula in Eq. (2) is derived from the area under a linear extrapolation from F_n to F_{n+1} based on F_n and F_{n-1}.

From Z transform theory it can be shown that the numerical integration formula in general has a transfer function for sinusoidal inputs which takes the form [1]

$$H_I^*(e^{j\omega h}) \equiv \frac{1}{j\omega[1 + e_I(j\omega h)^k]}\quad, \quad \omega h << 1 \qquad (4)$$

where for AB-2 integration, $k = 2$ and $e_I = 5/12$. The term $e_I(j\omega h)^k$ represents the error in the integrator transfer function compared with the ideal continuous integrator transfer function, $1/j\omega$. Based on the integrator model of Eq. (4) the transfer function gain and phase error in simulating any order of dynamic system, when quasi-linearized, can easily be obtained [1]. It can also be shown that the fractional error in any characteristic root as a result of integration truncation errors is given by

$$e_\lambda = \frac{\lambda^* - \lambda}{\lambda} \cong -\,e_I\,(\lambda h)^k\,, \quad |\lambda h| << 1 \qquad (5)$$

where λ is the continuous system root and λ^* is the equivalent root for the digital simulation. Thus it is apparent that the order k and error coefficient e_I for any given integration algorithm can be used to predict the dynamic errors that will be introduced into a simulation because of the finite step size h when using that algorithm. It should be noted that this methodology of error analysis is not applicable to multiple-pass integration methods, such as the Runge-Kutta algorithms, where different orders of integration algorithms are used in the various derivative evaluation passes that constitute a single integration step. It is applicable to all of the real-time methods considered in this paper.

As indicated in Eq. (1), the state variable derivative F will in general depend on the state X. Since the AB-2 algorithm in Eq. (2) involves the past derivative F_{n-1}, the next state X_{n+1} will depend on the past state X_{n-1} as well as the current state X_n. For this reason the AB-2 method introduces one extraneous state per integration. The characteristic roots

*Professor of Aerospace Engineering
Associate Fellow, AIAA

corresponding to these extraneous states damp rapidly for small integration step sizes, in which case they do not contribute significant errors to the simulation. However, when the step size becomes large, the extraneous roots can cause instability. In particular, for a negative real root λ, instability occurs when $\lambda h < -1$. This means that when AB-2 integration is used, the step size must be kept less than the shortest time constant in the system being simulated. Stability charts in the complex λh plane show the allowable step sizes when the roots of the continuous system are complex [2].

Consider next the Adams-Moulton two-pass predictor-corrector algorithm. Here the AB-2 method is used on the first pass to compute an estimate, X'_{n+1}, for the n+1 state. From this state and the input U_{n+1} the estimated derivative F'_{n+1} is in turn calculated. The corrector pass then computes X_{n+1} using F_n and F'_{n+1} with the following formula:

$$X_{n+1} = X_n + .5h\,(F'_{n+1} + F_n) \qquad (6)$$

If the estimate F'_{n+1} in Eq. (6) is repaced by the true derivative F_{n+1}, the formula represents implicit trapezoidal integration. It turns out that the explicit AM-2 method has the same asymptotic error coefficient, $e_I = -1/12$, as the implicit trapezoidal integration which it approximates. The order $k = 2$.

It should be noted that the AM-2 algorithm requires two passes through the state equations per integration step. At the beginning of the second pass, the input U_{n+1} is used in the calculation of the derivative estimate F'_{n+1}. But U_{n+1} will not be available in real time until the completion of the second pass. Hence AM-2 is not compatible with real-time inputs.

However, a second-order predictor-corrector algorithm suitable for real-time inputs can be constructed using the concept behind modified Euler integration. In modified Euler integration the state-variable derivative is represented at the midpoint of the integration step. Thus the integration formula becomes

$$X_{n+1} = X_n + h\,F_{n+1/2} \qquad (7)$$

If the derivative F is a function of the state X, which is normally the case, then it is necessary to compute an estimate for the state $X_{n+1/2}$ in order to evaluate $F_{n+1/2}$. In real-time Runge-Kutta 2 this is accomplished using Euler integration with a step size of $h/2$. The resulting real-time RK-2 integrator error coefficient $e_I = 1/6$. In the second-order real-time predictor-corrector method, denoted here as RTAM-2, the estimate for $X_{n+1/2}$ is computed using a second-order predictor algorithm. This leads directly to the following difference equations [3]:

$$X'_{n+1/2} = X_n + h(\tfrac{5}{8}F_n - \tfrac{1}{8}F_{n-1}) \qquad (8)$$

$$X_{n+1} = X_n + hF'_{n+1/2} \qquad (9)$$

where

$$F'_{n+1/2} = F(X'_{n+1/2}, U_{n+1/2}) \qquad (10)$$

The predictor formula in Eq. (8) for $X_{n+1/2}$ is derived from the area under a linear extrapolation based on F_n and F_{n-1}. The RTAM-2 given by Eqs. (8), (9) and (10) requires the input U_n at the start of the first pass and $U_{n+1/2}$ at the start of the second pass, both compatible with real time. Here the RTAM-2 integrator error coefficient $e_I = 1/24$, compared with -1/12 for standard AM-2. For a negative real root λ, instability occurs when $\lambda h < -2$. In the complex λh plane the overall stability boundary is slightly larger than the stability boundary for standard AM-2.

Thus the modified AM-2 integration denoted here as RTAM-2 can be used as a general, real-time integration method for simulating nonlinear systems. For small step sizes h it is twice as accurate as either implicit trapezoidal integration or standard AM-2 integration, neither of which are compatible with real-time inputs. It is ten times more accurate than AB-2 integration. It should be realized, however, that the RTAM-2 considered here is a two pass per step method. Thus it will normally take about twice as long for computer execution as AB-2. When this speed differential is taken into account, the modified AM-2 still exhibits 2.5 times the dynamic accuracy of AB-2 based on the approximate asymptotic formulas for small step size. In a real-time simula-tion the intermediate state $X'_{n+1/2}$ can be used as a real-time output which has full second-order predictor accuracy for the step size $h/2$. Use of both $X'_{n+1/2}$ and X_{n+1}, then, provides outputs at the sample rate of a single-pass method, even though a two-pass method is being utilized. Note also that the method uses two input samples per frame, U_n and $U_{n+1/2}$.

The final integration method considered in this section is a single-pass version of the RTAM-2 algorithm described above. The method computes the state-variable derivative F only at integer frame times rather than both half-integer and integer frame times, as in Eqs. (8), (9) and (10). Values of the state X at half-integer frame times are computed using the modified-Euler algorithm, while values of X at integer frame times are computed using a second-order predictor. The difference equations are the following [3]:

$$X_{n+1/2} = X_{n-1/2} + hF'_n \qquad (11)$$

$$X'_{n+1} = X_{n+1/2} + h(\tfrac{7}{8}F'_n - \tfrac{3}{8}F'_{n-1}) \qquad (12)$$

where

$$F'_n = F(X'_n, U_n) \qquad (13)$$

The predictor formula in Eq. (12) is derived from a linear extrapolation based on F'_n and F'_{n-1}. The SPRTAM-2 given by Eqs. (11), (12) and (13) requires the input U_n at the start of the nth integration step, which is compatible with real time. From Z transform theory we can show that the SPRTAM-2 integrator error coefficient $e_I = 1/24$, the same as that obtained previously for the two-pass RTAM-2. Since it executes twice as fast (one pass per integration step versus two for RTAM-2) and the dynamic errors are proportional to h^2, it will in general be four times as accurate.

The price we pay for the accuracy increase in the single-pass predictor-corrector method is reduced stability. For a negative real root λ, instability occurs for $\lambda h < -4/7$, compared with $\lambda h < -2$ for RTAM-2 (actually, $\lambda h < -1$ if one takes into account the doubled execution time for the two-pass RTAM-2). In the complex λh plane the overall stability boundary for SPRTAM-2 is therefore somewhat smaller than the boundary for either standard AM-2 or the RTAM-2 introduced here. It is also somewhat smaller than the stabilty boundary for AB-2 integration. In any event we should remember that the integrator error coefficient e_I in Eq. (4) and the characteristic root error in Eq. (5) are both based on approximate formulas that assume the integration step size is small in comparison with the reciprocal frequencies or eigenvalues, respectively. For the moderate step sizes normally used in flight simulation, final comparison of integration methods should be based on actual example simulations, as we shall see in a following section.

3. The Modified Euler Integration Method

Application of the modified-Euler integration method to the nonlinear flight equations can be understood by considering the following two vector state equations for the velocity vector V and the displacement vector D:

$$\dot{V} = A[D,V,U(\mathrm{t})]\ , \quad \dot{D} = V \tag{14}$$

To apply the modified-Euler method we represent the discrete velocity state V at half-integer frame times, denoted by $V_{n-1/2}$. The acceleration A and discrete displacement state D are represented at integer frame times, denoted by A_n and D_n, respectively. Then the modified-Euler difference equations become

$$V_{n+1/2} = V_{n-1/2} + hA_n\ , \quad D_{n+1} = D_n + hV_{n+1/2} \tag{15}$$

where

$$A_n = A(D_n, V'_n, U_n) \tag{16}$$

Here V'_n represents an estimate of V at the nth frame. To obtain this estimate we resort to the same predictor formula used above in Eq. (12) for the SPRTAM-2 method. Thus we let

$$V'_n = V_{n-1/2} + h(\tfrac{7}{8}A_{n-1} - \tfrac{3}{8}A_{n-2}) \tag{17}$$

Unlike the SPRTAM-2 method, however, we note here that the displacement state D_n in Eq. (16) is not a predictor-derived estimate but results directly from the modified-Euler integration algorithm itself in Eq. (15). This results in considerably improved stability, particularly for dynamic systems having quasi-linear charcteristic roots that are near the imaginary axis. In fact if A does not depend on V and depends linearly on D, as would be the case for the pure imaginary roots of an undamped dynamic system, it is easy to show that the modified-Euler method presented here exhibits exactly zero damping, regardless of the integration step size h [4].

We note that the predictor formula can be used to compute the displacement $D'_{n+3/2}$ in accordance with the formula

$$D'_{n+3/2} = D_n + h(\tfrac{7}{8}V_{n+1/2} - \tfrac{3}{8}V_{n-1/2}) \tag{18}$$

Thus at the end of the nth integration frame in a real-time simulation we can output the displacement state D_{n+1} as needed in real time and a prediction of the state, $D'_{n+3/2}$, one half step ahead of real time. This could be quite advantageous in compensating for other delays in an overall real-time simulation, such as the half-frame delay associated with the dynamics of zero-order DAC (digital-to-analog) extrapolators.

The nonlinear dependence of the acceleration A on the velocity V in Eq. (14) can often be expressed in terms of $V\partial A/\partial V$, where $\partial A/\partial V$ is not a function of V, or at worst is only slightly dependent on V. For example if A represents dQ/dt, the time derivative of pitch rate Q in the flight equations, then $\partial A/\partial Q$ is proportional to the aerodynamic stability dericvative C_{MQ}, i.e., the dimensionless pitching moment due to dimensionless pitch rate. C_{MQ} is normally independent of Q, although it may be dependent on other variables such as Mach number. Also, the overall $\partial A/\partial Q$ in this case will be independent of Q. Letting V be a scalar which represents the angular velocity Q, we can rewrite Eq. (14) as follows:

$$\dot{V} = C_0[D,U(t)] + C_1[D,U(t)]V \tag{19}$$

where $C_0 + C_1V = A$ and $C_1 = \partial A/\partial V$. Now, when mechanizing the modified-Euler difference equations (15) and (16) we can compute V'_n, the estimate of V at the nth frame, by the formula

$$V'_n = \tfrac{1}{2}(V_{n+1/2} + V_{n-1/2}) \tag{20}$$

From Eqs. (19) and (20) the difference equation (15) then becomes

$$V_{n-1/2} = V_{n-1/2} + h[C_0(D_n,U_n) + C_1(D_n,U_n)\frac{V_{n+1/2}+V_{n-1/2}}{2}] \tag{21}$$

With respect to the velocity state V this equation clearly represents implicit trapezoidal integration. However it can be solved to obtain the following explicit formula for $V_{n+1/2}$:

$$V_{n+1/2} = \frac{(1+hC_1/2)V_{n-1/2} + hC_0}{1-hC_1/2} \tag{22}$$

This formulation, i.e., the use of trapezoidal integration for the damping term, expands very substantially the stability region in the λh plane compared with the use of the predictor formula of Eq. (17) for the damping term [5]. It can also reduce appreciably the dynamic errors following transient inputs. The extra required computation is modest and consists mainly of an additional division.

In deriving Eq. (22) we have assumed that V ia a scalar, whereas V will in general be a vector. In this case $\partial A/\partial V$ will be a matrix, which must be inverted to obtain the explicit formula for $V_{n+1/2}$. Fortunately, the critical terms in this matrix in the case of the flight equations are the diagonal terms, in which case simple formulas similar to Eq. (22) involving only the diagonal terms can be derived. In particular, if P, Q, and R represent angular velocity components in roll, pitch, and yaw, respectively, difference equations similar to (22) can be written for $P_{n+1/2}$, $Q_{n+1/2}$, and $R_{n+1/2}$ where C_1 in each equation is proportional to the stability derivatives $C_{\ell P}$, C_{MQ}, and C_{NR} respectively.

4. ExampleSolutions of Flight Equations

In this section we compare the performance of the various real-time integration algorithms described in the previous two sections in the solution of actual flight equations. Since the largest characteristic roots for the rigid airframe are normally those associated with the short period pitching motion, we will only consider symmetric flight, i.e., the longitudinal equations of motion, in our example simulation. The conclusions regarding dynamic errors can safely extrapolated to the full six-degree-of-freedom case. The scalar rotational flight equations are invariably written with respect to aircraft body axes. However, the translational equations can be written with respect to either body or flight path axes [6]. Here we will use flight path axes, since they seem to be somewhat more suitable for the modified-Euler algotithm. In this case the longitudinal equations of motion can be written as follows:

$$\dot{V}_p = \frac{F_{wx}}{mV_p} \tag{23}$$

$$\dot{\alpha} = Q + \frac{F_{wz}}{mV_p} \tag{24}$$

$$\dot{Q} = \frac{M}{I_{yy}} \tag{25}$$

$$\dot{\Theta} = Q \tag{26}$$

$$\dot{H} = V_p \sin(\Theta - \alpha) \tag{27}$$

Here V_p is the total aircraft velocity, α is the angle of attack, Q is the pitch rate, Θ is the pitch angle, and H is the altitude; F_{wx} and F_{wz} are the external force components along the x and z flight-path axes, respectively, and M is the moment about the y body axis; finally, m and I_{yy} represent, respectively, the aircraft mass and pitch-axis moment of inertia. The following formulas were used to represent the external forces and moment:

$$F_{wx} = -qS\left(C_{D_0} + C_{DC_L^2} C_L^2\right) - g\sin(\Theta - \alpha) + \frac{T}{m}\cos\alpha \tag{28}$$

$$F_{wz} = -qS\left(C_L + C_{L_{\delta_e}}\delta_e\right) + g\cos(\Theta - \alpha) - \frac{T}{m}\sin\alpha \tag{29}$$

$$M = qcS\left(C_{M_0} + C_{M_\alpha}\alpha + C_{M_Q}\frac{c}{2V_p}Q + C_{M_{\dot{\alpha}}}\frac{c}{2V_p}\dot{\alpha} + C_{M_{\delta_e}}\delta_e\right) \tag{30}$$

where

$$q = \text{dynamic pressure} = \frac{1}{2}\rho V_p^2 \tag{31}$$

and

$$C_L = \text{lift coefficient} = C_{L_0} + C_{L_\alpha}\alpha \tag{32}$$

In these equations S is the aircraft wing area, g is the gravity acceleration, T is powerplant thrust, δ_e is elevator displacement, and c is the mean aerodynamic chord. The various C's represent aerodynamic coefficients and stability derivatives in accordance with the subscripts. In a full flight-envelope simulation these will be nonlinear functions of other variables such as V_p (through Mach number dependence), α, δ_e, and h. The actual difference equations used to solve (23) through (32) using modified Euler integration are presented in the Appendix.

As a specific example we consider a business jet flying at 40,000 feet at a speed of Mach 0.7 [7]. For the above flight condition the undamped natural frequency of the short-period mode is about 3 rad/sec and the damping ratio is 0.4. We consider the aircraft response to two different input functions. One is a step elevator displacement of -0.01 radians at the initial time $t = 0$. The second is the input function shown in Figure 1, which is a step elevator displacement with a one second rise time. Use of this input function tends to reduce the large transient errors caused by step inputs when predictor integration algorithms are used. It is also probably more representative of an actual pilot input. Figures 2 and 3 show the aircraft pitch rate and pitch angle response, as generated by the simulation using RK-4 integration with a step size $h = 0.05$ seconds. With this step size the RK-4 simulation is sufficiently accurate to serve as an ideal solution against which the real-time second-order algorithms described in this paper can be checked.

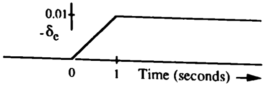


Figure 1. Finite rise-time elevator step input.

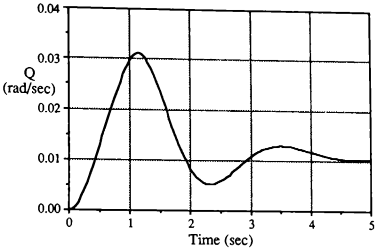


Figure 2. Pitch-rate response to input of Figure 1.

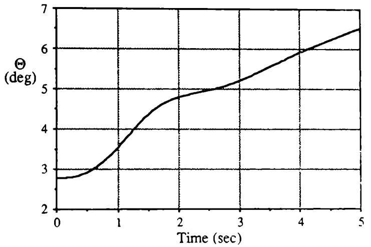


Figure 3. Pitch angle response to input of Figure 1.

The predictor algorithms considered in this paper all depend on the past as well as the present value of each state-variable derivative. This causes an initial startup ambiguity, since at the initial time t = 0 the past states and hence their derivatives are not known. They could be computed numerically prior to initiation of the simulation by integrating backwards one step using, for example, an RK algorithm. However, the usual method in real time simulation is to employ Euler simulation for the first step. Unfortunately, this can also cause startup transients which can mask the dynamic errors we are looking for in the second-order algorithms considered here. To circumvent this problem we have chosen to use a real-time RK algorithm for the first integration step. This consists of an Euler half-step followed by an RK-2 full step which uses the derivative as computed from the half-step result [1]. Subsequent integration steps use the particular second-order method being studied. In Figure 4 the error in pitch angle is plotted versus time with data points from the simulation using AB-2, RTAM-2, SPRTAM-2 and modified Euler integration, as described in this paper. For each algorithm the step size h =

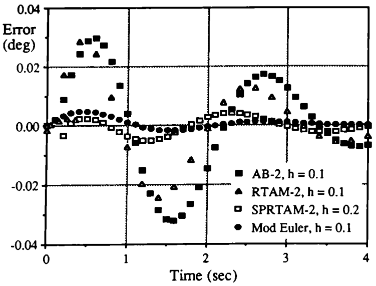


Figure 4. Pitch angle error versus time for step input in elev. displacement.

0.1 (10 integration steps per second), except that in the case of the RTAM-2 method we have used h = 0.2. This is because the RTAM-2 is a two-pass method which requires approximately twice the processor time per overall integration step in comparison with single-pass methods. From the figure it is evident that the modified-Euler algorithm performs slightly better than the SPRTAM-2 algorithm, with the AB-2 and RTAM-2 algorithms exhibiting considerably larger errors. For smaller step sizes the RTAM-2 method becomes significantly more accurate than AB-2, finally approaching an asymptotic limit equal to 0.4 times the AB-2 error, as noted earlier in Section 2. For smaller step sizes the SPRTAM-2 and modified-Euler methods continue to show an order of magnitude advantage over AB-2.

Next we consider the finite rise-time step input. In this case, in order to have the example be representative of an ongoing simulation, we have delayed three integration steps before applying the elevator input function of Figure 1. In Figure 5 the error in pitch angle is plotted versus time with data points from the simulation using AB-2, SPRTAM-2 and modified Euler integration. Again the step size h = 0.1. Once more the superior accuracy of the modified-Euler method is apparent. It is interesting to note the transient error introduced not only by the input slope discontinuity at t = 0.3, but also the additional transient error introduced by the second input slope discontinuity at t = 1.3.

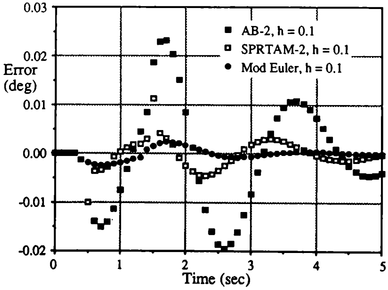


Figure 5. Pitch angle error for the input function of Figure 1, delayed in time by 1 second.

In this paper we have only considered second-order integraion algorithms. This is because it is usually not worthwhile to consider higher-order methods for the moderate dynamic accuracy (the order of one percent) normally considered adequate for real-time flight simulation. Higher order methods sutable for real-time simulation also tend to be less stable.

5. Conclusions

We have shown that the modified form of Euler integration described in this paper, when applied to the six-degree-of-freedom flight equations, gives results that are substantially more accurate than the AB-2 method which is usually employed. This has been demonstrated by considering the asymptotic formulas for characteristic root and transfer function errors, and by comparing actual time-domain errors in the response to control surface input functions. Following a transient input, the modified-Euler method produces an output response in the next integration step, whereas AB-2 integration exhibits an additional one-step delay in displacement response. The inherent nature of the modified-Euler algorithm permits it to produce outputs at half-integer steps. This feature can be used to produce an accurate half-frame lead in real-time output displacement. The modified Euler method may also be useful in the integration of state and costate equations in real-time mechanization of Kalman filters for navigation and control systems.

We have also shown that two other second-order integration methods, a real-time AM-2 predictor-corrector and a single-pass version of the same, represent more accurate alternatives to AB-2 integration for flight simulation.

Appendix

The flight equations used for the numerical results presented in this paper are given by Eqs. (23) through (32). Application of the AB-2, RTAM-2 and SPRTAM-2 integration algorithms to these equations is straightforward. Application of the modified Euler method requires additional explanation, which is the purpose of this appendix. When the six-degree-of-freedom flight equations are represented entirely in body axes, the six velocity states consist of U, V, W, the components of airframe velocity, and P, Q, R, the components of airframe angular velocity, along the body axes, x, y, z, respectively. The six displacement states consist of three position coordinates, normally latitude, longitude, and altitude, and, when Euler angles are used, three angular position coordinates, Φ, Θ, and Ψ, . Alternatively, four quaternions, e_1, e_2, e_3, and e_4, can be used as angular position states, from which direction cosines and Euler angles can be computed. To prevent accumulation of errors due to the the redundant fourth quaternion state, constraint terms which maintain $e_1^2 + e_2^2 + e_3^2 + e_4^2 = 1$ are added to the right side of each quaternion state equation.

When the translational equations of motion for the six-degree-of-freedom flight equations are derived using flight-path axes, the velocity states are represented by V_p, the total aircraft velocity, α, the angle of attack, and β, the sideslip angle [6]. When only the nonlinear symmetric (longitudinal) equations of motion are considered, the state equations become (23) through (27), where the equation for horizontal position has not been included. Based on the way in which modified-Euler integration was introduced in Section 3, the velocity states V_p, α and Q would be represented at half-integer frames, with the position states Θ and H represented at integer frames. For the nth integration frame this results in the computation of the n+1/2 velocity state from the n-1/2 velocity state, followed by computation of the n+1 position state from the n position state using the n+1/2 velocity state just obtained. However, from Eq. (24) it is apparent that it would be better to represent the angle of attack α at integrer frames, even though it is derived from a velocity state equation. This is because the dominant term on the right side of Eq. (24) affecting the high-speed dynamics is the pitch-rate Q, which is represented at half-integer frames. The other term in Eq. (24), F_{wz}/mV_p, is the negative of the flight-path-axis pitch rate, and is generally much smaller in magnitude than Q. This is the reason for representing α at integer frames in the modified Euler flight equations. When the lateral equations of motion are considered in a full six-degree-of-freedom simulation, the same rationale is used to represent the velocity state β at integer rather than half-integer frames when using modified Euler integration.

With this as background, we now list the actual difference equations used in each modified-Euler integration step when solving the longitudinal flight equations.

Variables available at the start of the nth integration frame:

$$V_{p_{n-1/2}},\ V'_{p_n},\ Q_{n-1/2},\ Q'_n,\ \alpha_n,\ \alpha'_{n+1/2},\ \Theta_n,\ \Theta'_{n+1/2},\ H_n,\ H'_{n+1/2},\ \delta_{e_n},$$
$$\delta_{e_{n-1}},\ \dot{V}_{p_{n-1}},\ \dot{Q}_{n-1},\ (F_{wz}/mV_p)_{n-1/2}$$

Difference equations for the nth integration frame (in order of execution):

$$q_n = .5\rho_n V'^{2}_{p_n},\ \ C_{L_n} = C_{L_0} + C_{L_\alpha}\alpha_n,\ \ C_{L_{n+1/2}} = C_{L_0} + C_{L_\alpha}\alpha'_{n+1/2} \tag{A.1}$$

$$\dot{V}_{p_n} = -[(q_n S/m_n)(C_{D_0} + C_{D_{C_L^2}} C^2_{L_n}) - g\sin(\Theta_n - \alpha_n) + (T_n/m_n)\cos(\alpha_n)]/V'_{p_n} \tag{A.2}$$

$$Q_n = q_n S c/I_{yy_n}[C_{M_0} + C_{M_\alpha}\alpha_n + (.5c/V'_{p_n})((C_{M_Q} + C_{M_{\dot\alpha}})Q_n + C_{M_{\dot\alpha}}(1.5(F_{wz}/mV_p)_{n-1/2} - .5(F_{wz}/V_p)_{n-3/2}) + C_{M_{\delta_e}}\delta_{e_n}] \tag{A.3}$$

$$V_{p_{n+1/2}} = V_{p_{n-1/2}} + h\dot{V}_{p_n} \tag{A.4}$$

$$V'_{p_{n+1}} = V_{p_{n+1/2}} + h(.875\dot{V}_{p_n} - .375\dot{V}_{p_{n-1}}) \tag{A.5}$$

$$Q_{n+1/2} = Q_{n-1/2} + h\dot{Q}_n \tag{A.6}$$

$$Q'_{n+1} = Q_{n+1/2} + h(.875\dot{Q}_n - .375\dot{Q}_{n-1}) \tag{A.7}$$

$$(F_{wz}/mV_p)_{n+1/2} = -q_n(S/m_n)[C_{L_{n+1/2}} + C_{L_{\delta_e}}(1.5\delta_{e_n} - .5\delta_{e_{n-1}})] + g\cos(\Theta'_{n+1/2} - \alpha'_{n+1/2}) - (T_n/m_n)\sin(\alpha'_{n+1/2}) \tag{A.8}$$

$$\alpha_{n+1} = \alpha_n + h[Q_{n+1/2} + (F_{wz}/V_p)_{n+1/2}] \tag{A.9}$$

$$\alpha'_{n+3/2} = \alpha_{n+1} + h\{.875[Q_{n+1/2} + (F_{wz}/V_p)_{n+1/2}] - .375[Q_{n-1/2} + (F_{wz}/V_p)_{n-1/2}]\} \tag{A.10}$$

$$\Theta_{n+1} = \Theta_n + hQ_{n+1/2} \tag{A.11}$$

$$\Theta'_{n+3/2} = \Theta_{n+1} + h(.875Q_{n+1/2} - .375Q_{n-1/2}) \tag{A.12}$$

$$H_{n+1} = H_n + hV_{p_{n+1/2}}\sin[.5(\Theta_{n+1} + \Theta_n) - .5(\alpha_{n+1} + \alpha_n)] \tag{A.13}$$

Some comments regarding the above equations are in order. As noted previously, in an actual full flight-envelope simulation the aerodynamic coefficients which appear as constants in the equations will in fact be nonlinear functions of variables such as angle of attack, Mach number, control-surface dispacement, etc. The calculation of these multivariable functions by table lookup and linear interpolation usually constitutes a sizeable fraction of the total processor time for one integration step.

Eq. (A.1) indicates the necessity of computing both C_{L_n} and $C_{L_{n+1/2}}$. In a full simulation the drag coefficient in Eq. (A.2) would probably be computed as a nonlinear function of Mach number, angle of attack, and perhaps other variables such as flap position and elevator position. Typically, the lift coefficient would involve a nonlinear function of the same variables. Note that the $\dot\alpha$ term in Eq. (30) is synthesized in Eq. (A.3) in terms of Q and F_{wz}/mV_p in accordance with the formula in Eq. (24). The $\dot\alpha$ term actually results from a first-order approximation to the time delay associated with the effect of wing downwash on the horizontal tail. An alternative method for simualing this is to include a pure time delay proportional to $1/Vp$ for the fraction of the $C_{M_\alpha}\alpha$ term due to the downwash effect.

In Eq. (A.8) the dynamic pressure q_n has been used rather than $q_{n+1/2}$, even though the right side of Eq. (A.8) in general is represented at the n+1/2 frame. This is because both the density ρ and velocity V_p vary slowly enough that it seems hardly worthwhile to compute the dynamic pressure at both the n and n+1/2 frame, although this could be done with a modest additional amount of calculation. Also, in Eqs. (29) and (A.8) we have not included aerodynamic lift terms involving Q and $\dot\alpha$ simply because their effect generally turns out to be negligible.

In the full six-degree-of-freedom flight equations the counterpart of Eq. (26), or its equivalent difference equation (A.11), would be nonlinear state equations for the three Euler angle rates. Or, if quaternions are used, there would be four state equations invloving e_1, e_2, e_3, and e_4, as well as P, Q, and R. The corresponding difference equations would compute $e_{1_{n+1}}$, $e_{2_{n+1}}$, $e_{3_{n+1}}$, and $e_{4_{n+1}}$ from formulas involving $P_{n+1/2}$, $Q_{n+1/2}$, and $R_{n+1/2}$. To convert body axis velocity components to earth-axis velocity components at the n+1/2 frame, direction cosines are used. These are computed at the n+1/2 frame from quaternions at the n+1/2 frame which in turn are computed as the average value of the quaternions at the n and n+1 frame. Finally, the earth-axis position coordinates are computed at the n+1 frame with modified Euler integration using the earth axis velocity components at the n+1/2 frame as the input derivatives. Eqs. (A.11) and (A.13) represent the equivalent to these calculations in our simplified longitudinal case.

References

1. Howe, R.M., "Transfer Function and Characteristic Root Errors for Fixed-Step Integration Algorithms," *Transactions of the Society for Computer Simulation*, 2 (4): 293-320.

2. Benyon, P.K., "A Review of Numerical Methods for Simulation," *Simulation*, 11(5):219-238.

3. Howe, R.M., "The Use of Real-time Predictor-corrector Integration for Flight Simulation," *Proc. of the SCS Simulation Conference Simulators V*, Orlando, 18-21 April, 1988, pp 38-42.

4. Howe, R.M., "Simulation of Linear Systems Using Modified Euler Integration Methods," *Transactions of the Society for Computer Simulation*, 5 (2): 125-152

5. Howe, R.M., "A Performance Comparison of Integration Algorithms in Simulating Flexible Structures," *Proc. of the NASA Workshop on Computational Aspects in the Control of Flexible Systems*, July 12-14, 1988 Williamsburg, VA.

6. Fogarty, L.E., and R.M. Howe, "Computer Mechanization of Six-Degree-of-Freedom Flight Equations," *Simulation*, 11(4): 187-193.

7. Roskam, J., "*Airplane Flight Dynamics and Automatic Flight Controls,*" Vol. 1, 1982, pp. 616-624, Roskam Aviation and Engineering Corporation, Route 4, Box 274, Ottawa, Kansas 66067.

A NEW METHOD FOR OPTIMALLY FITTING Z TRANSFORMS TO LAPLACE TRANSFORMS 89-3307-CP

Mark S. Fineberg
McDonnell Aircraft Company
A Division Of
McDonnell Douglas Corporation

Abstract

There are often requirements to simulate a known Laplace transfer function on a digital computer. The method most commonly seen is the bilinear transformation (which is frequently called "Tustin") and a refinement called prewarping. Another possibility is the Reddy method. A new method is presented which has proven superior in all respects tested.

A brief summary of the derivation of the older methods is given using the Z transform notation.

A new method is presented based on a general form of the Z transform which is set equal to the desired Laplace transform. Replacing Z by its power series expansion in S yields several equations in S. When these equations are solved, values of the coefficients of the Z transform are obtained as functions of the coefficients of the Laplace transform.

Four sets of formulas are given, first order open (predictor),first order closed (corrector), second order open, and second order closed. The first order forms are fairly simple but the second order forms are more complex.

The response using the new method is compared to the response of older methods. In each case the new method is shown superior.

The derivation of coefficients is shown in the appendix for each of the four forms discussed.

Introduction

In digital simulation a common requirement is to generate a difference equation corresponding to a known Laplace transfer function. The method most commonly seen is the bilinear transformation[1,2] (which is frequently called "Tustin") and a refinement called prewarping. Another possibility is to use the Reddy Method[3]. This paper introduces a new method and shows it to be superior the older methods.

A difference equation of the form relevant to this discussion may be conveniently represented by the Z-transform[4] where $Z = e^{St}$.

The bilinear transformation can be derived (or intuitively pointed to) in several ways. My favorite is to start with the series

$$\ln(x) = 2\left(y + \frac{y^3}{3} + \frac{y^5}{5} + \cdots \right) \quad \text{where}$$

$$y = \frac{x-1}{x+1}$$

and throw away all higher order terms. That leaves

$$\ln(x) = \frac{2(x-1)}{x+1}\ .$$

Let $x = Z$,

$$\ln(Z) = St = \frac{2(Z-1)}{Z+1} \qquad \text{so}$$

$$S = \frac{2(Z-1)}{t(Z+1)}\ .$$

Substituting for S in the Laplace transform produces the bilinear Z transform approximation.

We can force the bilinear transform to fit exactly at any particular critical frequency (ω) by using $2\tan(\frac{\omega t}{2})/\omega$ for t. This is known as prewarping.

The Reddy method fits a polynomial in time to the input and finds what the continuous response would be to that input over the interval. The value at the end of the interval is the approximation to the output. Any convenient order may be chosen for the polynomial, higher orders make for more complex formulae. Either open or closed intervals may be used. Reddy's method is derived entirely in the time domain but the results may be easily expressed with transform notation.

New Method

The new formulae are based on a general form of the Z transform such as

$$\frac{A + BZ^{-1} + CZ^{-2}}{D + EZ^{-1} + FZ^{-2}}$$

and setting it equal to the Laplace transform. Then replace Z with its power series expansion in S,

$$Z = 1 + ST + \frac{(ST)^2}{2!} + \frac{(ST)^3}{3!} + \cdots ,$$

which yields an equation in powers of S. Next collect like powers of S into separate equations until there are as many equations as unknown coefficients. When the equations are solved, values of the coefficients of the Z transform are obtained as functions of the coefficients of the Laplace transform.

The Z transform does not have to be in exactly the form shown above. It may be of first order. Predictive forms may also be derived by using a factor of Z^{-1}. Multiplying the Z transform by Z^{-1} is, in effect, asking for a fit of the Laplace transform advanced by one frame to compensate for a one frame computational delay. (Better terminology would be to note that the usual form is on a *closed* interval and the "predictor" is on an *open* interval.)

When this procedure is followed for a first order over first order in S and Z the formula derived is exactly the same as the standard Tustin! There is some extra power in that open (predictive) forms can be derived in addition to the usual closed versions. It is also comforting to note that in the limit as ω approaches zero the formulae derived are the same as the classical integrators.

The second order form used is a first order over a second order. This is all that is needed since the second order over second order may be formulated using partial fractions.

New Formulae

The following definitions are convenient:

$X = ST$;
$P = \tau_1/T$;
$M = \tau_2/T$;
$N = \tau_3/T^2$.

FIRST ORDER CLOSED FORM

(Same as Tustin !)

$$\frac{1 + \tau_1 S}{1 + \tau_2 S} \approx \frac{A + BZ^{-1}}{C + DZ^{-1}}$$

$A = 1 + 2P$

$B = 1 - 2P$

$C = 1 + 2M$

$D = 1 - 2M$

FIRST ORDER OPEN FORM

$$\frac{1 + \tau_1 S}{1 + \tau_2 S} \approx \frac{Z^{-1}(A + BZ^{-1})}{C + DZ^{-1}}$$

$A = 2 - 3M + 3P - 2MP + 2P^2$

$B = M - P + 2MP - 2P^2$

$C = M - P - 2M^2 + 2MP$

$D = 2 - 3M + 3P + 2M^2 - 2MP$

SECOND ORDER CLOSED FORM

$$\frac{1 + \tau_1 S}{1 + \tau_2 S + \tau_3 S^2} \approx \frac{A + BZ^{-1} + CZ^{-2}}{D + GZ^{-1} + HZ^{-2}}$$

$$\begin{aligned}
A = & - M^2 + 3MN + 2MP + 6N^2 - 3NP - P^2 - 3M^2P \\
& - 6MNP + 6MP^2 + 36N^2P + 6NP^2 - 3P^3 \\
& - 36MNP^2 + 36NP^3
\end{aligned}$$

$B = - 4M^2 + 8MP + 60N^2 - 4P^2 - 60MNP + 60NP^2$

$$\begin{aligned}
C = & - M^2 - 3MN + 2MP + 6N^2 + 3NP - P^2 + 3M^2P \\
& - 6MNP - 6MP^2 - 36N^2P + 6NP^2 + 3P^3 \\
& + 36MNP^2 - 36NP^3
\end{aligned}$$

$$\begin{aligned}
D = & - M^2 + 3MN + 2MP + 6N^2 - 3NP - P^2 - 3M^3 \\
& + 6M^2P + 36MN^2 - 6MNP - 3MP^2 + 72N^3 + 6NP^2 \\
& - 36M^2NP - 72MN^2P + 36MNP^2 + 72N^2P^2
\end{aligned}$$

$G = A + B + C - D - H$

$$\begin{aligned}
H = & - M^2 - 3MN + 2MP + 6N^2 + 3NP - P^2 + 3M^3 \\
& - 6M^2P - 36MN^2 - 6MNP + 3MP^2 + 72N^3 \\
& + 6NP^2 + 36M^2NP - 72MN^2P \\
& - 36MNP^2 + 72N^2P^2
\end{aligned}$$

SECOND ORDER OPEN FORM

$$\frac{1 + \tau_1 S}{1 + \tau_2 S + \tau_3 S^2} \approx \frac{Z^{-1}(A + BZ^{-1} + CZ^{-2})}{D + GZ^{-1} + HZ^{-2}}$$

$$\begin{aligned}
A = & 6M - 6N - 6P - 17M^2 + 63MN + 40MP - 78N^2 \\
& - 27NP - 23P^2 - 63M^2P + 150MNP + 90MP^2 \\
& - 108N^2P - 78NP^2 - 27P^3 - 72M^2P^2 \\
& + 108MNP^2 + 72MP^3 - 108NP^3
\end{aligned}$$

$$\begin{aligned}
B = & 6M + 6N - 6P - 14M^2 + 12MN + 22MP + 12N^2 \\
& + 24NP - 8P^2 - 12M^2P - 84MNP - 12MP^2 \\
& + 144N^2P + 12NP^2 + 24P^3 + 72M^2P^2 \\
& - 144MNP^2 - 72MP^3 + 144NP^3
\end{aligned}$$

$$\begin{aligned}
C = & M^2 - 3MN - 2MP - 6N^2 + 3NP + P^2 + 3M^2P \\
& + 6MNP - 6MP^2 - 36N^2P - 6NP^2 + 3P^3 \\
& + 36MNP^2 - 36NP^3
\end{aligned}$$

$$\begin{aligned}
D = & M^2 + 3MN - 2MP - 6N^2 - 3NP + P^2 - 3M^3 \\
& + 6M^2P + 36MN^2 + 6MNP - 3MP^2 - 72N^3 - 6NP^2 \\
& - 36M^2NP + 72MN^2P + 36MNP^2 - 72N^2P^2
\end{aligned}$$

$$\begin{aligned}
G = & 6M - 6N - 6P - 8M^2 + 60MN + 22MP - 60N^2 \\
& - 24NP - 14P^2 - 24M^3 + 72M^2N + 12M^2P \\
& - 144MN^2 + 60MNP + 12MP^2 + 144N^3 \\
& - 60NP^2 - 72M^3P + 144M^2NP + 72M^2P^2 \\
& - 144MN^2P - 144MNP^2 + 144N^2P^2
\end{aligned}$$

$H = A + B + C - D - G$

The detailed derivation ot these formulae is shown in the appendix below.

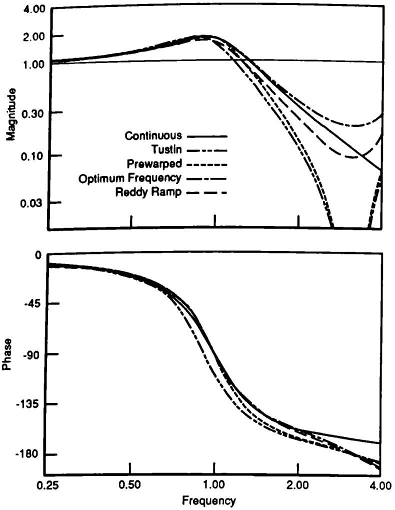
Results

The nominal case which has been used to show the test results is a second order transfer function with a 1 radian per second resonant frequency and a damping factor of 0.3 sampled at a frequency of 1 Hertz. Five plots are shown in each graph: the continuous solution, normal Tustin, prewarped Tustin, the Reddy Ramp, and this recommended method. Figure 1 shows the magnitude and phase of the exact answer together with the approximations*. There is no name for the new method so it is refered to as "optimum frequency" because it uses coefficients which optimize the response in the frequency domain. The new method is shown to extend the accuracy of the Tustin for about one extra octave. The prewarped Tustin is, as it has to be, exactly right at the critical frequency, ω, but it pays a penalty between zero and ω.

Since the new method is derived in the frequency domain it is not surprising that its frequency response is good but how well does it fit in the time domain? Figure 2 shows the response to a step input. It reveals that the same conclusions can be drawn in both the time and frequency domains and that the new method is superior to its competitors.

---

* This is similar in form to Fig. 3.6a in Reference [1].



GP91-0040-12-A/MAS

Figure 1. Nominal Case

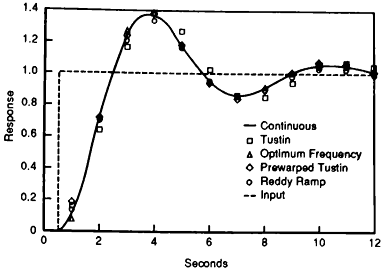
Figures 3, 4, 5, and 6 bracket the nominal case showing what happens at lower and higher resonant frequencies and more or less well damped systems. The expected behavior is shown, again the new method is superior.

Figure 7 shows the behavior of the predictive form. The Tustin plot shows what happens when a frame delay is ignored. The magnitude is acceptable but the phase is terrible. One way to improve the phase is to insert a predictor. A trapezodial predictor was combined with a Tustin and it does improve the phase but the amplitude error is unacceptable. The second order form of the new method was used and it proved a very good fit almost out to the Nyquist frequency. However great care must be used in any predicting situation, in some examples very poor fits were unexpectedly found.

There is one situation where the new method should not be used. That is where the additional complexity of computing the coefficients poses too great a burden and Tustin is accurate enough. In simulation we most commonly compute the coefficients once during initialization and use them during each pass. However, ocasionally we need to dynamically update time constants. This situation needs to be studied on a case by case basis.

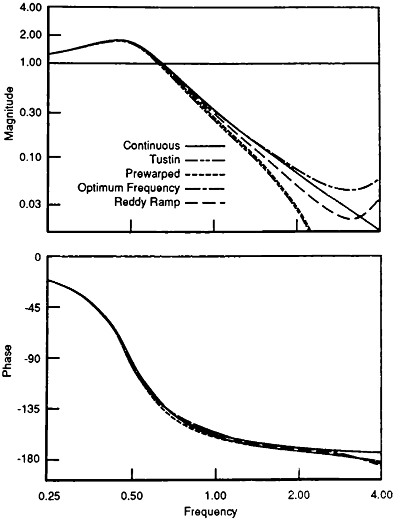
CONCLUSION

A new method for digitally representing Laplace transfer functions has been presented. It has been shown to have less error than those in common use.



GP91-0040-13-A/MAS

Figure 2. Step Response



GP91-0040-14-A/MAS

Figure 3. Half Frequency

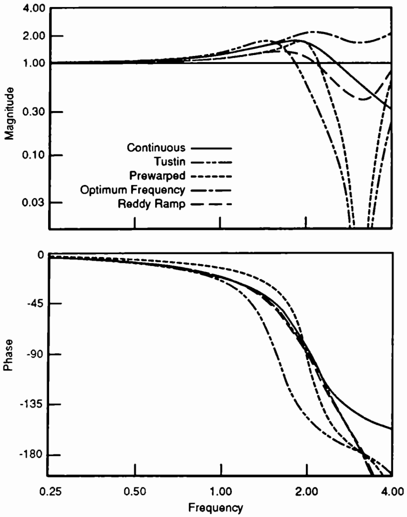
References

1. Franklin, G., and Powell, J., Digital Control of Dynamic Systems, Addison Wesley, 1980

2. Stanley W., Dougherty G., and Dougherty R., Digital Signal Processing 2nd Ed., Reston, 1984

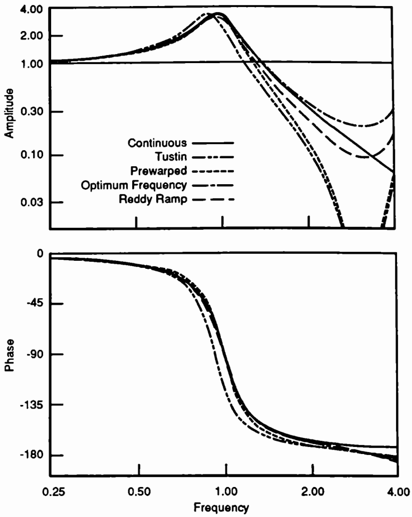
3. Rosamond, D., "Stepwise Calculation of Responses of First and Second Order Linear Systems", McDonnell Guidance and Control Mechanics Note No. 184, 1965

4. Truxall, J. G., Automatic Feedback Control Systems Synthesis, McGraw-Hill, 1955



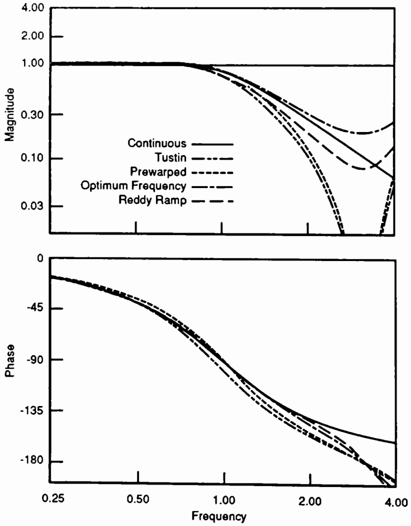
GP91-0040-15-A/MAS

Figure 4. Double Frequency



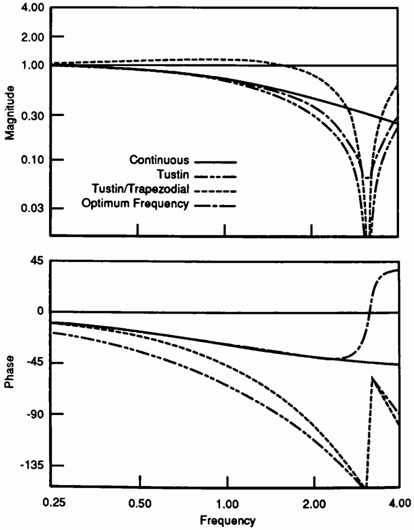
GP91-0040-17-A/MAS

Figure 6. Half Damping



GP91-0040-16-A/MAS

Figure 5. Double Damping



GP91-0040-18-A/MAS

Figure 7. Predictors

Appendix

The formulae shown in the NEW FORMULAE section are derived in this appendix.

Let $X = ST$.

Then, since $Z = e^{ST}$

$Z = 1 + X + \frac{X^2}{2!} + \frac{X^3}{3!} + \cdots ,\qquad Z^{-1} = 1 - X + \frac{X^2}{2!} - \frac{X^3}{3!} + \cdots ,$

and $Z^2 = 1 + 2X + 2X^2 + \frac{4}{3}X^3 + \cdots .$

$P = \tau_1/T,\ M = \tau_2/T,\ \&\ N = \tau_3/T^2$ so that

$\tau_1 S = PX,\ \tau_2 S = MX$, and $\tau_3 S^2 = NX^2$.

FIRST ORDER CLOSED FORM

$\frac{1 + \tau_1 S}{1 + \tau_2 S} \approx \frac{A + BZ^{-1}}{C + DZ^{-1}}$ or dividing by C,

$\frac{1 + \tau_1 S}{1 + \tau_2 S} \approx \frac{A' + B'Z^{-1}}{1 + D'Z^{-1}}$

$(1 + MX)(A'Z + B') = (1 + PX)(Z + D')$

$(A' + A'MX)Z + B'MX + B' = (1 + PX)Z + D' + D'PX$

$(A' - 1 + (A'M - P)X)Z + (B'M - D'P)X + B' - D' = 0$

$(A' - 1 + (A'M - P)X)(1 + X + X^2/2) + (B'M - D'P)X + B' - D' = 0$

$X^0 :\quad A' - 1 + B' - D' = 0$

$X^1 :\quad A' - 1 + A'M - P + B'M - D'P = 0$

$X^2 :\quad A'M - P + \frac{A' - 1}{2} = 0$

$2A'M - 2P + A' - 1 = 0$

$A' (1 + 2M) = 1 + 2P$

$A' = \frac{1 + 2P}{1 + 2M}$

$A' - 1 + A'M - P + (D' + 1 - A') M - D'P = 0$

$A' + A'M - A'M - 1 - P + M + D' (M - P) = 0$

$\frac{1 + 2P}{1 + 2M} - 1 - P + M = D' (P - M)$

$1 + 2P + (M - P - 1) (1 + 2M) = D' (P - M) (1 + 2M)$

$1 + 2P + M - P - 1 + 2M^2 - 2MP - 2M = D' (P - M) (1 + 2M)$

$P - M - 2MP + 2M^2 = D' (P - M) (1 + 2M)$

$P - M - 2M (P - M) = D' (P - M) (1 + 2M)$

$1 - 2M = D' (1 + 2M)$

$D' = \frac{1 - 2M}{1 + 2M}$

$B' = D' + 1 - A' = \frac{1 - 2M}{1 + 2M} + \frac{1 + 2M}{1 + 2M} - \frac{1 + 2P}{1 + 2M}$

$B' = \frac{1 - 2P}{1 + 2M}$

Multiplying by $1 + 2M$

$A = 1 + 2P$

$B = 1 - 2P$

$C = 1 + 2M$

$D = 1 - 2M$

Which is as shown in the NEW FORMULAE section.

FIRST ORDER OPEN FORM

$$\frac{1 + \tau_1 S}{1 + \tau_2 S} \approx \frac{Z^{-1}(A + BZ^{-1})}{C + DZ^{-1}}$$ or dividing by C,

$$\frac{1 + \tau_1 S}{1 + \tau_2 S} \approx \frac{Z^{-1}(A' + B'Z^{-1})}{1 + D'Z^{-1}}$$

$$(1 + PX)(Z + D') = (1 + MX)(A' + B'Z^{-1})$$

$$(1 + PX)Z + D' + D'PX = A' + A'MX + (B' + B'MX)Z^{-1}$$

$$(1 + PX)Z = A' - D' + (A'M - D'P)X + (B' + B'MX)Z^{-1}$$

$$(1 + PX)(1 + X + \frac{X^2}{2}) = A' - D' + (A'M - D'P)X + (B' + B'MX)(1 - X + \frac{X^2}{2})$$

$$1 + (P + 1)X + (P + \frac{1}{2})X^2 = A' + B' - D' + (A'M - D'P - B' + B'M)X + (\frac{B'}{2} - B'M)X^2$$

X^0 : $1 = A' + B' - D'$

X^1 : $P + 1 = A'M - D'P - B' + B'M$

X^2 : $P + \frac{1}{2} = \frac{B'}{2} - B'M$

$$(1 - 2M)B' = 1 + 2P$$

$$B' = \frac{1 + 2P}{1 - 2M}$$

$$P + 1 = A'M - (A' + B' - 1)P + B' (M - 1)$$

$$A' (M - P) = 1 + P - P + B' (1 - M + P) = 1 + \frac{1 + 2P}{1 - 2M} (1 - M + P)$$

$$A' = \frac{1 - 2M + 1 - M + P + 2P - 2MP + 2P^2}{(1 - 2M)(M - P)}$$

$$A' = \frac{2 + 3P - 3M - 2MP + 2P^2}{(1 - 2M)(M - P)}$$

$$D' = A' + B' - 1 = \frac{2 + 3P -3M + 2P^2 - 2MP + (M - P)(1 + 2P)}{(1 - 2M)(M - P)} - 1$$

$$D' = \frac{2 + 3P - 3M + 2P^2 - 2MP + M - P + 2MP - 2P^2 - (1 - 2M)(M - P)}{(1 - 2M)(M - P)}$$

$$D' = \frac{2 + 2P - 2M - M + P + 2M^2 - 2MP}{(1 - 2M)(M - P)} = \frac{2 + 3P - 3M - 2MP + 2M^2}{(1 - 2M)(M - P)}$$

Multiplying by (1 - 2M)(M - P)

$A = 2 - 3M + 3P - 2MP + 2P^2$

$B = M - P + 2MP - 2P^2$

$C = M - P - 2M^2 + 2MP$

$D = 2 - 3M + 3P + 2M^2 - 2MP$

These are also as shown as NEW FORMULAE.

SECOND ORDER CLOSED FORM

$$\frac{1 + \tau_1 S}{1 + \tau_2 S + \tau_3 S^2} \approx \frac{A + BZ^{-1} + CZ^{-2}}{D + GZ^{-1} + HZ^{-2}}$$ or dividing by D,

$$\frac{1 + \tau_3 S}{1 + \tau_1 S + \tau_2 S^2} \approx \frac{A' + B'Z^{-1} + C'Z^{-2}}{1 + G'Z^{-1} + H'Z^{-2}}$$

$$(1 + PX)(Z + G' + H'Z^{-1}) = (1 + MX + NX^2)(A'Z + B' + C'Z^{-1})$$

$$(1 + PX)(1 + X + \frac{X^2}{2} + \frac{X^3}{6} + \frac{X^4}{24} + G' + H' - H'X + \frac{H'X^2}{2} - \frac{H'X^3}{6} + \frac{H'X^4}{24}) =$$

$$(1 + MX + NX^2)(A' + A'X + \frac{A'X^2}{2} + \frac{A'X^3}{6} + \frac{A'X^4}{24} + B' + C' - C'X + \frac{C'X^2}{2} - \frac{C'X^3}{6} + \frac{CX^4}{24})$$

$$X^0 : \quad 1 + G' + H' = A' + B' + C'$$

$$X^1 : \quad 1 - H' + P + G'P + H'P = A' - C' + A'M + B'M + C'M$$

$$X^2 : \quad \frac{1}{2} + \frac{H'}{2} + P - H'P = \frac{A'}{2} + \frac{C'}{2} + A'M - C'M + A'N + B'N + C'N$$

$$X^3 : \quad \frac{1}{6} - \frac{H'}{6} - \frac{P}{2} + \frac{H'P}{2} = \frac{A'}{6} - \frac{C'}{6} + \frac{A'M}{2} + \frac{C'M}{2} + A'N - C'N$$

$$X^4 : \quad \frac{1}{24} + \frac{H'}{24} + \frac{P}{6} - \frac{H'P}{6} = \frac{A'}{24} + \frac{C'}{24} + \frac{A'M}{6} - \frac{C'M}{6} + \frac{A'N}{2} + \frac{C'N}{2}$$

$$1 + P + P(A' + B' + C' - H' - 1) - H' + H'P = A' + A'M + B'M - C' + C'M$$

$$(1 + M - P)A' + (M - P)B' - (1 - M + P)C' + H' = 1$$

$$(1 + 2M + 2N)A' + 2NB' + (1 - 2M + 2N)C' - (1 - 2P)H' = 1 + 2P$$

$$(1 + 3M + 6N)A' - (1 - 3M + 6N)C' + (1 - 3P)H' = 1 + 3P$$

$$(1 + 4M + 12N)A' + (1 - 4M + 12N)C' - (1 - 4P)H' = 1 + 4P$$

At this point the problem is, in principle, solved since there are five equations in five unknowns. However, in practice the solution is quite onerous as well as very error prone. To alleviate these problems a program was written to evaluate a fourth order determinant when all the coefficients are linear in three constants. (One variable in the original set can be easily substituted out.) This program was used to solve these equations for A′, B′, C′, and H′ as a function of M, N, and P. Even if it is difficult to solve, it is quite easy to verify that a solution is correct. Several sets of numerical values for M, N, and P were substituted into the solution and the equations were indeed satisfied, giving good confidence that errors have not crept into the solution.

These results, after multiplication by the denominator, have been presented as A, B, C, D, G, and H in the NEW FORMULAE section.

SECOND ORDER OPEN FORM

$$\frac{1 + \tau_3 S}{1 + \tau_1 S + \tau_2 S^2} \approx \frac{Z^{-1}(A + BZ^{-1} + CZ^{-2})}{D + GZ^{-1} + HZ^{-2}} \quad \text{or dividing by D,}$$

$$\frac{1 + \tau_3 S}{1 + \tau_1 S + \tau_2 S^2} \approx \frac{Z^{-1}(A' + B'Z^{-1} + C'Z^{-2})}{1 + G'Z^{-1} + H'Z^{-2}}$$

$$(1 + PX)(Z^2 + G'Z + H') = (1 + MX + NX^2)(A'Z + B' + C'Z^{-1})$$

$$(1 + PX)Z^2 = (A' - G' + (A'M - G'P)X + A'NX^2)Z + B' - H' + (B'M - H'P)X + B'NX^2 + (C' + C'MX + C'NX^2)Z^{-1}$$

$$(1 + PX)(1 + 2X + 2X^2 + \tfrac{4}{3}X^3 + \tfrac{2}{3}X^4) = (A' - G' + (A'M - G'P)X + A'NX^2)(1 + X + \frac{X^2}{2} + \frac{X^3}{6} + \frac{X^4}{24}) + B' - H' + (B'M - H'P)X + B'NX^2 + (C' + C'MX + C'NX^2)(1 - X + \frac{X^2}{2} - \frac{X^3}{6} + \frac{X^4}{24})$$

$$X^0 : \quad 1 = A' - G' + B' - H' + C'$$

$$X^1 : \quad 2 + P = A' - G' + A'M - G'P + B'M - H'P - C' + C'M$$

$$X^2 : \quad 2 + 2P = \frac{A' - G'}{2} + A'M - G'P + A'N + B'N + \frac{C'}{2} - C'M + C'N$$

$$X^3 : \quad \frac{4}{3} + 2P = \frac{A' - G'}{6} + \frac{A'M - G'P}{2} + A'N - \frac{C'}{6} + \frac{C'M}{2} - C'N$$

$$X^4 : \quad \frac{2}{3} + \frac{4}{3}P = \frac{A' - G'}{24} + \frac{A'M - G'P}{6} + \frac{A'N}{2} + \frac{C'}{24} - \frac{C'M}{6} + \frac{C'N}{2}$$

$$2 + P = A' - G' + A'M - G'P + B'M - (A' + B' + C' - 1 - G')P - C' + C'M$$

$$(1 + M - P)A' + (M - P)B' - (1 - M + P)C' - G' = 2$$

$$(1 + 2M + 2N)A' + 2NB' + (1 - 2M + 2N)C' - (1 + 2P)G' = 4(1 + P)$$

$$(1 + 3M + 6N)A' - (1 - 3M + 6N)C' - (1 + 3P)G' = 4(2 + 3P)$$

$$(1 + 4M + 12N)A' + (1 - 4M + 12N)C' - (1 + 4P)G' = 16(1 + 2P)$$

This form was also solved using the process described for the closed form and the results are in the NEW FORMULAE section.

89-3308-CP

NONLINEAR MODEL FOLLOWING CONTROL APPLICATION TO A FLIGHT SIMULATOR CONTROL LOADER

Wayne C. Durham*
Virginia Polytechnic Institute and State University

Abstract

Perfect explicit model following control laws for a flight simulator control loader problem are developed. The control loader is digitally controlled and uses a direct drive DC motor (torque motor) to simulate airplane control stick forces and displacements. The control laws deal directly with the nonlinearities present in the modeled control system and in the motor, and reduce the problem to one of a second order regulator involving the error. The error dynamics for the nonlinear problem are determined by conventional pole placement methods. The method is illustrated with an example, wherein the simulated control system features breakout forces, nonlinear friction, and hysteresis in the spring forces.

Nomenclature

J	=	Stick plus motor polar moment of inertia
T_P	=	Max motor torque
T_R	=	Motor ripple torque
ω_R	=	Motor ripple frequency
K_V	=	Motor viscous damping term
T_F	=	Motor static friction term
K_p	=	Motor constant
i_p	=	Motor current ($K_p i_p$ = motor torque)
m_p	=	Mass of stick, modeled as a point mass
L_p	=	Length of stick
T_p	=	Applied torque
θ	=	Stick deflection
x_p	=	Plant state variable vector
u_p	=	Plant control
x_m	=	Model state variable vector
u	=	Model Control
e	=	Error vector
K_e	=	Error feedback gain matrix

Problem Description. The design problem is that of a digitally controlled flight simulator control loader system. The flight simulator control stick is driven by an electric torque motor, which opposes or assists the pilot's applied forces on the stick in a way that makes the stick forces and motions simulate that of a real flight control system. The simulated system is to contain various nonlinearities, including freeplay about the trim position, breakout forces, hysterisis, and a nonlinear variation of friction with displacement.

Perfect Model Following Control.

Consider a plant and model whose linearized equations of motion are given by:

$$\dot{x}_p = \begin{bmatrix}\dot{x}_p^1\\ \dot{x}_p^2\end{bmatrix} = \begin{bmatrix}A^1\\ A_p^2\end{bmatrix}\begin{bmatrix}x_p^1\\ x_p^2\end{bmatrix} + \begin{bmatrix}0\\ I\end{bmatrix}u_p \qquad (1)$$

$$\dot{x}_m = \begin{bmatrix}\dot{x}_m^1\\ \dot{x}_m^2\end{bmatrix} = \begin{bmatrix}A^1\\ A_m^2\end{bmatrix}\begin{bmatrix}x_m^1\\ x_m^2\end{bmatrix} + \begin{bmatrix}0\\ B_m^2\end{bmatrix}u \qquad (2)$$

The systems have these characteristics:

(1) The plant and the model are of the same order;

(2) For a plant with m controls, there are exactly m plant equations that depend on these controls, and these controls appear through the identity matrix for those m equations.

(3) For the n-m plant equations that do not directly depend on the controls, there are at least n-m model equations that do not depend on the model's controls. These n-m equations are identical with the corresponding plant equations (the submatrix A^1 is the same for both plant and model).

(4) The remaining m equations are arbitrary, including B_m^2. The externally applied control vector u is of any order less than or equal to m.

* Graduate Student

These equations are said to be in the standard form for perfect model following. It is easily shown that equations in this form satisfy all sufficiency tests for perfect model following.

From the standard form, the control law for perfect, error correcting model following may be shown to be:

$$u_p^* = \dot{x}_m^2 + K_e e - A_p^2 x_p \qquad (3)$$

with corresponding error dynamics

$$\dot{e} = \begin{bmatrix} \dot{e}^1 \\ \dot{e}^2 \end{bmatrix} = \begin{bmatrix} A^1 \\ -K_e \end{bmatrix} \begin{bmatrix} e^1 \\ e^2 \end{bmatrix} \qquad (4)$$

Now consider the plant and model equations in a more general form, where both state vectors are arranged in a form similar to that in (1) and (2). We will require that the state equations not explicitly containing controls be linear differential equations, but place no such restriction on the explicitly controlled states:

$$\begin{aligned} \dot{x}_p^1 &= A^1 x_p \\ \dot{x}_p^2 &= f_p(x_p) + u_p \end{aligned} \qquad (5)$$

$$\begin{aligned} \dot{x}_m^1 &= A^1 x_m \\ \dot{x}_m^2 &= f_m(x_m, u) \end{aligned} \qquad (6)$$

Here $f_p(\cdot)$ and $f_m(\cdot)$ are (different) nonlinear functions, and the plant control is addititive. Then (5) may be solved directly for the control, and the control law is

$$u_p^* = \dot{x}_m^2 + K_e e - f_p(x_p) \qquad (7)$$

Substituting this control into the plant equations (5) yields the same results for error dynamics as was given in (4). It does not matter that $f_p(\cdot)$ represents a nonlinear function: it is only necessary that it can be computed for given values of the plant states. Also, it is clear that $f_m(\cdot)$ does not have to be a linear function, only that the model state rates can be computed for given system inputs.

Nonlinear systems that can be cast in the form of (5) can thus be made to follow nonlinear models in the form of (6).

Application

Figure 1 is a schematic of the system in the longitudinal axis. With modifications, the following discussion applies to lateral and directional control loaders as well.

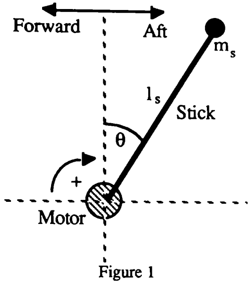


Figure 1
System Schematic

Equations of Motion

The plant equations of motion are

$$J\ddot{\theta}_p = m_p g L_p \sin\theta_p + T_A - K_V \dot{\theta}_p \pm T_F + K_p i_p \qquad (8)$$

In (8), the sign of the static friction term (T_F) is taken to oppose the direction of motion. The last term in (8) reflects the assumption that the motor develops torque instantaneously with changes in current. Expressed as state variables, this equation is in the standard form for perfect model following.

With $x_P^1 \equiv \theta_P$, $u_P \equiv \frac{K_P}{J} i_P$:

$$\begin{aligned} \dot{x}_p^1 &= x_p^2 \\ \dot{x}_p^2 &= f_p(x_p) + u_p \end{aligned} \qquad (9)$$

where

$$f_p = \frac{1}{J}[m_p g L_p \sin x_P^1 + T_A - K_V x_P^2 \pm T_F] \qquad (10)$$

The control system being simulated is described by a similar second order equation of the form

$$\ddot{\theta}_m = f_m(\theta_m, \dot{\theta}_m, T_A) \tag{11}$$

This equation represents the breakout, friction, viscous damping, etc. present in the control system being simulated.

With $x_m^1 \equiv \theta_m$, $u \equiv T_A$:

$$\dot{x}_m^1 = x_m^2$$

$$\dot{x}_m^2 = f_m(x_m^1, x_m^2, u) \tag{12}$$

The control law is given by (7), and the error dynamics by (4), with

$$A^1 = [0 \;\; 1] \tag{13}$$

and

$$K_e = [k_1 \;\; k_2] \tag{14}$$

The gain matrix elements are picked to place the poles of the error dynamics as desired.

Measurement errors

Assume that the physical characteristics of the system are known and remain constant during operation. The measured or estimated values of the states and of applied torque are denoted by variables with a circumflex (^) mark, and the differences between these and the actual values are denoted with tilde (~) marks (e.g., $\hat{x}_p = x_p + \tilde{x}_p$). The control being applied to the plant is given by

$$u_p^* = \dot{x}_m^2 + K_e(x_m - \hat{x}_p) - f_P(\hat{x}_P) \tag{15}$$

This leads to

$$\dot{x}_p^2 = \dot{x}_m^2 + f_P(x_P) - f_P(\hat{x}_P) + K_e(x_m - \hat{x}_p) \tag{16}$$

Assuming small errors in measurement and estimation, $f_P(\hat{x}_P)$ may be linearized about the exact values of the variables, resulting in:

$$\dot{x}_p^2 = \dot{x}_m^2 + K_e e + \frac{1}{J}\tilde{T}_A - K^*\tilde{x}_p \tag{17}$$

where

$$K^* \equiv K_e - \left[\frac{m_p g L_p \cos x_p^1}{J} \quad \frac{-K_V}{J}\right] \tag{18}$$

The error dynamics now are given by:

$$\dot{e} = \begin{bmatrix} 0 & 1 \\ -k_1 & -k_2 \end{bmatrix} e + \begin{bmatrix} 0 \\ K^* \end{bmatrix} \tilde{x}_p - \begin{bmatrix} 0 \\ 1 \end{bmatrix} \frac{\tilde{T}_A}{J} \tag{19}$$

The original error dynamics are thus modified by the addition of exogenous inputs of the form shown.

Commutation Ripple

Because commutation in the motor is done in discrete steps, there is a small variation in average torque during rotation of the armature. This ripple may be modeled as a variation in K_P with θ_P. Figure 2 (greatly exaggerated) shows the effect of ripple on torque for a given current input.

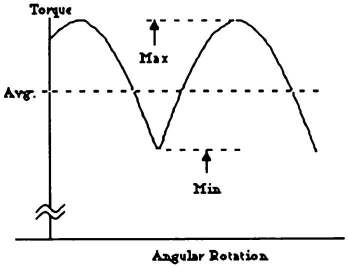


Figure 2
Commutation Ripple

Ripple is specified as a percentage deviation (T_R) about the average torque, where the average is the arithmetic average of maximum and minimum torque for a given current input. Define K_{PO} as the nominal value of K_P and the ripple function $f_R(x_P^1)$ $(=f_R(\theta_P))$ such that

$$K_P(x_P^1) = K_{PO}[1 + f_R(x_P^1)] \quad (20)$$

Applying this to the error dynamics yields

$$\dot{e}_2 = -Ke - f_R(x_P^1)u_P \quad (21)$$

where u_P is determined using the nominal value of K_P.

Analysis of the effect of ripple on the error dynamics is complicated by the fact that $f_R(t)$ depends on the rate of stick deflection. The error feedback gains may be made large to minimize the ripple effect, but this is at the expense of noisy tracking.

Alternatively, if sufficiently accurate measurements of θ_P are available, the control may be calculated using an approximation to the actual value of K_P instead of K_{PO}. This will change the form of the forcing function in (21) and, if the approximation is good, will reduce its magnitude.

Example

The parameters of the torque motor considered for this application are as follows:

T_P	60 lb-ft
K_P	3.11 lb-ft/Amp
T_F	0.83 lb-ft
K_V	8.0 lb-ft-sec/rad
T_R	4 %
ω_R	97 cycles/rev
J_{motor}	0.041 lb-ft-sec^2
Peak Current	19.3 Amps

The control stick was taken as having a length of 1.5 feet and a mass of 0.2 slug. Commutation ripple was modeled as a rectified sine wave:

$$f_R = T_R\left(2\left|\sin(\omega_R\theta_P)\right| - 1\right) \quad (22)$$

This function was applied to the input to the motor. In order to simulate the fact that the ripple function is not known exactly, an approximation to (22) was used in calcualating the control laws. This approximation consisted of a linear interpolation of a table of seven values of the function given by (22). The error dynamics resulting from this approximation are given by (21), where the forcing function involves the difference between the actual and approximated ripple functions:

$$\dot{e}_2 = -Ke - [f_R(x_P^1)_{Actual} - f_R(x_P^1)_{Approx}]u_P \quad (23)$$

For the simulation results shown, a constant measurement bias error of +0.02 radian was assumed for θ_P, and of 0.02 radian per second for the angle rate. A +2% error in the measurement of applied torque was also assumed. The effect of this is to further modify the error dynamics, first by affecting the accuracy of the approximation given in (23), and second by the factors given in (19):

$$\dot{e}_2 = -Ke - [f_R(x_P^1)_{Actual} - f_R(\hat{x}_P^1)_{Approx}]u_P + \left(k_1 - \frac{m_P g L_p \cos x_P^1}{J}\right)\tilde{x}_P^1 - \left(k_2 + \frac{K_V}{J}\right)\tilde{x}_P^2 - \frac{\tilde{T}_A}{J} \quad (24)$$

where $\tilde{x}_P^1 = \tilde{x}_P^2 = 0.02$, and $\tilde{T}_A = 0.02T_A$.

Equation (24) is difficult to analyze, primarily because of the motor ripple term. It is comprised of an unknown part, $f_R(x_P^1)_{Actual}$, and an estimate based on possibly erroneous measurement, $f_R(\hat{x}_P^1)_{Approx}$. Increased gains will not generally reduce the effects of bias errors in measurements of angle and angle rates, since these terms are linear in k_1 and k_2. In this simulation, values of k_1 and k_2 were determined by trial and error.

Model Characteristics. The control stick characteristics to be simulated were modeled as follows:

Static friction: a nonlinear function of stick position, varying from five ft-lbs at one radian forward displacement to four ft-lbs at one radian aft displacement, with values less than 1.5 ft-lbs within 0.5 radian of centered.

Spring constant: 30.0 ft-lbs per radian, with hysteresis of ± 0.05 radian.

Breakout: ± 2.0 ft-lbs.

Viscous damping: 30.0 ft-lb-sec per radian.

Dead space: ± 2.0 ft-lbs about the centered position.

The resulting system was simulated using fourth order Runge-Kutta integration with a fixed step size of 0.025 second. Initial conditions for the simulations were taken as zero. The control input was a series of ramp functions, as shown in figure 3. The acceleration response to this input of the modeled control stick is shown in figure 4.

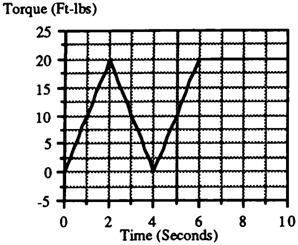


Figure 3
Control History

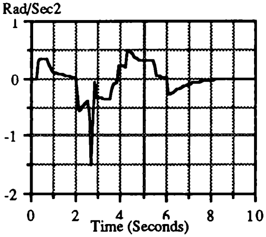


Figure 4
Model Acceleration

Results

For gains of $k_1 = 1.0$, $k_2 = 2.0$, angle following errors on the order of 0.2 radian were seen. Gains were increased to $k_1 = 25.0$, $k_2 = 5.0$, with results as shown in figures 5 and 6.

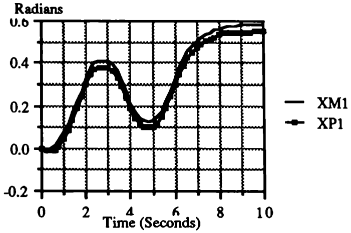


Figure 5
Stick Position

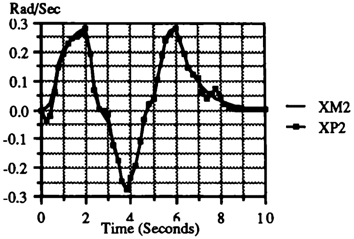


Figure 6
Stick Rate

The power supply requirements for this case are shown in figure 7.

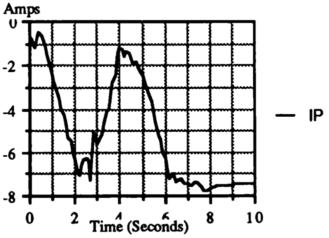


Figure 7
Current Requirements

Simulation results indicated that ripple effects may be reduced by a combination of higher feedback gains, and by "dividing them out" using an approximation based on the measured angle. For the motor characteristics chosen, ripple will go through a complete cycle each 0.065 radian of stick travel. Unless the measured value of θ_P is near this value, any scheme to approximate the ripple function may be useless. The latter method requires a precision in measurement that is inversely proportional to the ripple frequency. This argues for selecting a motor with smaller values of ripple frequency, or for the use of gearing. Increased feedback gains are effective in eliminating both ripple and measurement errors. In the example presented above, satisfactory performance was attained without excessively high gains, and the current requirements remained well within the maximum allowed.

Conclusions

Within the scope of the example presented, model following control may be successfully applied to the design of a digital control loader using a direct drive DC motor. Its implementation requires the measurement of stick position and rate, and of the applied torque. On-line calculations are required to determine the moments developed by the motor in response to the measured parameters, and to integrate the equations of motion of the control stick being simulated.

The effects of motor ripple may be reduced by increasing the error feedback gains, by approximating the value of the ripple through accurate position measurement, or by a combination of both. Proportional bias errors in torque measurements, may be compensated for by increased error feedback gains, whereas constant bias errors in position and rate measurements will degrade the performance of the system for any selection of gains.

Model following control was shown to be insensitive to nonlinearities in the equations of motion of the control system being simulated. In the example presented, the modeled control system incorporated several nonlinearities often present in conventional flight control systems, and these were faithfully reproduced in the simulation results.

SIMULATION MADE EASY; THE DEVELOPMENT OF AN INTEGRATED DATA DRIVEN SIMULATION OPERATING SYSTEM

David A. Poole
Director, Advanced Technology
Veda Systems Division
California, Maryland

Abstract

The development of mathematic models can be a time consuming task for aerodynamic engineers, and the process can be difficult and expensive: changes to any part of the model or derivative data base require software support and a re-compile of the real time code. About three years ago we became involved in the development of a low cost simulator based on a distributed micro-processor architecture. During this process we had to develop an integrated operating system that would allow the user simple single point access to the many processors in the system. The result of our design efforts was the development of a complete simulation operating system that allows the user to fully define any 6 DOF aerodynamic models through the use of a simple menu system. The development of this integrated simulation operating system has changed many of our pre-conceived ideas about flight simulation. It allows mathematic models to be easily implemented with no special skills, and changes to existing models can be carried out quickly and without software support.

Background

We believe that the use of flight simulation is one of the key tools available to the aero RDTE engineer. As well as the usual benefits normally quoted, such as cost, time and safety, there is an additional, and important aspect; the very process of modeling an aircraft leads to a greater knowledge of the flight characteristics. Before discussing the data driven simulation operating system that we developed, it is worth considering the background that led to its development.

Most of the large research and training simulators are expensive, difficult to use, need long lead times, require an extensive engineering staff, and can be of unknown fidelity in some areas and are not set up for the rapidly changing test environment. We perceived a need for a simulator that would have none of these limitations and yet sacrifice nothing in fidelity in the areas of flying qualities and performance. With the availability of more capable microprocessors we decided to build a specialized model processor that would form part of an easy to use simulator and which would be available to our engineers and customers for the host of RDT&E tasks in which our company was involved. The final system that we developed has all these features, and has been fully implemented in our GENESIS line of Engineering Flight Simulators.

All aerodynamic mathematic models used in real time manned simulators use a similar coefficient based structure. The coefficients describe the forces and moments generated by the aerodynamics and the controls, and are summed, dimensioned, integrated and transformed in the same manner for all six degree of freedom models. These coefficients are usually stored as data points in look up tables, and can generally have up to three dimensions. Most simulators use a single, capable computer to process all the real time executable aerodynamic routines. The usual approach is to design and code a number of interrelated modules specifically for each aircraft and simulator combination. This has the result that generally the only people that can implement, understand or change the mathematic model are software engineers; the aerodynamic engineers are often unfamiliar with the complex machine-dependent real time software structures. This facet has led to a discontinuity between the users and the developers, and mathematic modeling has become something of a black art to much of the aero-engineering community.

Since some of the characteristics of the processing pipeline can have a good deal of impact on the validity of the overall simulation, we have discussed these factors below. They include the transport time, the computational accuracy, and the system bandwidth of the overall simulation.

Transport Time Considerations

An area of simulator performance that is extremely important is the total transport time of the complete system. Even the measurement of this time is not well addressed, and the term seems to mean different things to different people. A good deal of research has been applied to the subject, but not all of it is directly applicable to any particular simulator or flight task. It is generally accepted that for tactical aircraft, pilot workload increases and task performance is degraded when the overall transport time starts to exceed around 130 - 170 milliseconds. It seems obvious that the delay that is acceptable would depend on the dynamic characteristics of the aircraft being modeled and the pilot gain that is required to accomplish the task. The work that we carried out in this area shows some correlation between overall delay and the damped natural frequency of the aircraft at the conditions applicable to the task. We therefore set out a requirement for the overall system delay to not exceed the equivalent of between 15 and 30 degree phase delay at the highest damped natural frequency being modeled, and never be more than 120 milliseconds (Figure 1).

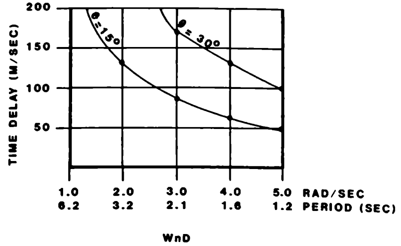


Figure 1
Transport Time Boundaries

The variation between 15 and 30 degrees depended on the pilot gain applied, and for high gain tasks such as gun tracking, was nearer the 15 degree boundary. The calculation or measurement of the system delay is also subject to various interpretations, from the inclusion of nothing but the model processing time to the incorporation of all the system individual block delays. Even this does not adequately address the problem from the perspective of relative effects on pilot performance.

The typical simulator will have a number of subsystems, such as the control modeling system and the display systems(s). Each of these often has a different iteration rate and processing time that must be considered in the computation of the overall delay. One problem in computing the total delay is that the pilot does not move the controls at the start of any frame; control input can occur at any time, and appears to total system as a sort of RMS function. The average time delay that the pilot feels is the total transport time of the system plus one half of any asynchronous frames; this is actually a zero order hold in digital system terminology, and it occurs at the input and at any non-synchronous interface within the computation pipeline. Since the A/D, mathematical model, and visual systems typically run at different processing rates, for very good reasons, the total transport time in this typical system will be the total of the individual frame times plus half the frame time of each separate processor. This total transport time is the parameter that must be used when deciding system performance factors (Figure 2).

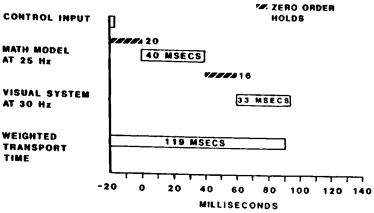


Figure 2
Zero Order Hold Delays

System Bandwidth Considerations

Another area that should be considered is the bandwidth that was required through the processing pipeline. The bandwidth of a digital system is related to the iteration rates of any processors in the pipeline, while the transport time is related to the sum of the different processing times as stated above. The response of a simulator with one ten Hertz processor is not the same as one with two synchronous twenty Hertz processors in serial. The transport times are the same in each case, but the maximum theoretical bandwidth is doubled in the second case. It seems to be generally accepted that the minimum bandwidth should exceed twenty Hertz for most simulation tasks.

System Accuracy Considerations

Any lack of computation accuracy appears to the pilot as the equivalent of mechanical freeplay in the system. We had no data on the practical requirements of this characteristic, so we decided to compute the mathematical model at a full 64/80 bits in order to avoid any problems in this area.

The GENESIS Model Processor

In the process of developing the system architecture, it was clear from the outset that we would have to use a number of fast microprocessors linked by a common bus structure. There are both positive and negative aspects to this type of architecture. The positive point is that the hardware can easily be changed to accommodate developments in processing capabilities, and that the total system throughput, in a well structured system, is not directly linked to bus bandwidth. The negative point is that each of the processors may well require a different operating system, have to be loaded with executive and data code at run time, and the input and output of the different processors linked in real time. This effectively means that a high level executive must be developed that insulates the user from these tasks, and forms a system shell between the user and the system processors.

In the distributed processor architecture that we developed the total computational task is split into functionally independent areas, with each functional element being hosted on a separate processor, or collection of processors, with only processed system states being passed on the main system bus. Each of these independent processors then has a separate dedicated bus to its own co-processors and real time memory. The net effect is to unload the main systems buses, and distribute the bus intensive computational tasks among the separate sub processor system buses. This characteristic is illustrated in Figure 3.

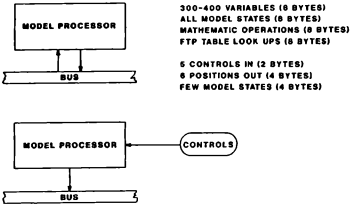

MODEL PROCESSOR
BUS
300-400 VARIABLES (8 BYTES)
ALL MODEL STATES (8 BYTES)
MATHEMATIC OPERATIONS (8 BYTES)
FTP TABLE LOOK UPS (8 BYTES)
5 CONTROLS IN (2 BYTES)
6 POSITIONS OUT (4 BYTES)
FEW MODEL STATES (4 BYTES)
MODEL PROCESSOR
CONTROLS
BUS

Figure 3
Bus Activity

In this figure we show a hypothetical distributed architecture system that uses a single independent processor for aerodynamic modeling; in actual practice this task could be split amongst a number of processors. In the example we show the three control inputs (roll, pitch and yaw) passing into a model processor; the outputs would be three positions in the world co-ordinate space and three attitude angles. Compare this with the effect of passing all the internal model states on the same bus; even a simple mathematic model may have several hundred aerodynamic co-efficients and additional derived model states.

Processor Architecture

A simplified diagram of the final GENESIS bus architecture is shown in Figure 4. It uses two main system buses, one for mathematic modeling and secondary displays, and one for the real world graphics system, linked by high speed reflected memory. These two buses each have a dedicated bus control processor, and are only used for data transfer and control duties. The distributed 32 bit micro-processors have additional parallel buses, each with a dedicated 32 bit bus, co-processors and memory. Typical processing speeds for these control processors vary from 1.5 to 8 MIPS, dependent on their task within the system.

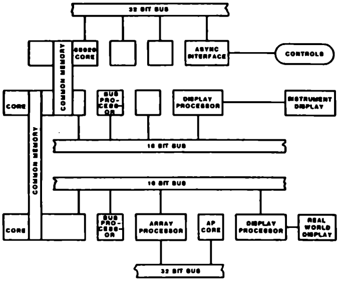


Figure 4
GENESIS Architecture

The Data Driven Operating System

As we already mentioned, one of the major problems in distributed microprocessor architecture is that user interface with, and control of, the total system becomes a complex task. We developed an executive to accomplish these functions; it loads the data files required and controls the real time operations. At this time it became clear that if the full utility of the system were to be realized, we would have to provide a method for users to input new models easily and rapidly without recourse to cumbersome operating system and compiler tasks. At this stage it is worth reviewing the structure of an aerodynamic model in order to clarify the various operations that are required.

Model Structure

A 6 DOF aerodynamic model consists of a number of elements, shown in Figure 5. After any control processing has been carried out, the control positions, generally as control surface angles, are passed to the model. At this stage the model can be logically divided into two paths; in these the control angles are multiplied by the non-dimensionalized coefficients. In the forces path the multipliers will be coefficients such as CYdr (sideforce due to rudder angle), and CLda (roll moment due to aileron angle) in the moments path. Lift and drag axis forces will be calculated in a similar manner. These moments and forces are then summed for each of the controls and dimensionalized. The remainder of the model will transform the forces and moments into the body axis, and calculate the accelerations by dividing the moments and forces by the correct inertia terms. The resultant accelerations will be integrated to produce rates, and transformed to the earth axis.

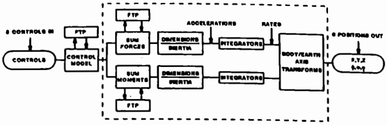


Figure 5
Aerodynamic Model

Most classical model structures contain all the code necessary to accomplish all of these functions; however the only part that actually changes between different aircraft models are the various non-dimensionalized coefficients used in the summing process. These generally vary with aerodynamic states such as Mach number, angle of attack and angle of sideslip. Due to the very non-linear nature of these coefficients, they are generally stored as multi-dimensional look up tables. For instance, the longitudinal control effect Cmde will usually vary with angle of attack and Mach number. In order to change the characteristics of the model, as many as several hundred of these look up tables have to be identified and modified. As might be expected there are additional problems; different aerodynamic models use different coefficients, dimensionalization terms, and may have a number of unique control surfaces. The result is that it becomes extremely difficult, even for experienced mathematic modelers, to change an existing model without experiencing difficulties in implementation. When the problems of compile time and configuration control are added the overall task can become fairly severe.

In order to avoid these difficulties, we developed a core aerodynamic model containing the equations of motion. This part of the model is fully generic, and is used whatever final aircraft is implemented.

The discrete integrators used in any model have a response analogous to an analogue integrator, in that they can affect system phase and gain. Many models in the past have used Euler integrators, but these can exhibit poor effects and some model frequencies. An examination of a number of different integrators showed that the second order Adams - Bashforth routines possessed many desirable qualities, and we used these in the core model. We carried out this type of analysis throughout the model structure, with the aim of developing a technically correct approach for our system. At this stage we had a core aerodynamic model with no method of inputting the desired coefficients. We solved this in two steps.

Coefficient Implementation

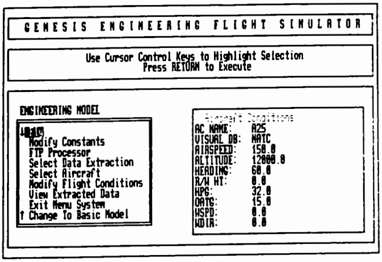
The first stage was to develop a look up table processor linked to a menu system, that allows the user to name any coefficient and define the associated dimensions and dimensionalization terms. We allowed these values to be input as ASCII data, and provided an utility to create the necessary run-time files. We structured the menu system into individual axis forces and moments, and provided a help utility to assist the user through the process. The actual format that we selected was in accordance with the NASA Function Table Processor data structure, so that the data tables from many existing military models could be transferred directly into the system. This model can accept over four hundred independent coefficients with up to four arguments or five dimensions, and we called it the Engineering Model.

The second stage was to provide a rapid prototyping facility so that point conditions could easily be modeled without having to input unnecessary data. It can be visualized that if the aerodynamic coefficients mentioned above are fixed, and not allowed to vary with parameters such as Mach and angle of attack, the final model will operate at a fixed set of conditions, although it will still be free to change airspeed, altitude etc, and will exhibit a full 6 degrees of freedom. We provided this capability for two reasons: the first was to allow the users to very rapidly create a new model for specialized flight tests, and the second was to provide an extremely effective teaching aid, since the implementation of a model in this fixed coefficient model structure is obvious to every flight test engineer and test pilot. We called this the Basic model.

The final tasks were to provide similar menus for environmental setup, the aircraft physical characteristics and simulation control. We also provided the capability of setting a number of associated parameters such as the processing speed, storing and loading models, and controlling peripheral system such as the real world visual displays. Any number of aerodynamic models can be developed and stored using this sytem, depending on storage media space.

Menu Structure

At the highest level, the user is presented with the menu shown in Figure 6.

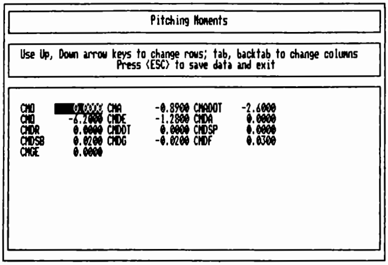


```
G E N E S I S   E N G I N E E R I N G   F L I G H T   S I M U L A T O R

          Use Cursor Control Keys to Highlight Selection
                      Press RETURN to Execute

ENGINEERING MODEL
↓[illegible]                              Aircraft Conditions
  Modify Constants                     AC NAME:     A25
  FTP Processor                        VISUAL DB:   NATC
  Select Data Extraction               AIRSPEED:    150.0
  Select Aircraft                      ALTITUDE:    12000.0
  Modify Flight Conditions             HEADING:     60.0
  View Extracted Data                  R/W HT:      0.0
  Exit Menu System                     HPG:         32.0
↑ Change To Basic Model                OATG:        15.0
                                       WSPD:        0.0
                                       WDIR:        0.0
```

Figure 6
System Control Menu

At this level the user can select any of the sub-menus associated with the development of the model, or choose the Engineering or Basic models. It generally takes several seconds to load a complex model into the various distributed processors, and the user then has all the usual control (stop, freeze, return to trim, etc) functions.

An example of the Basic model longitudinal moments is shown in Figure 7.



```
                         Pitching Moments

Use Up, Down arrow keys to change rows; tab, backtab to change columns
                   Press <ESC> to save data and exit

CM0        0.0000 CMA      -0.8900 CMADOT   -2.6000
CMQ       -6.2000 CMDE     -1.2800 CMDA      0.0000
CMDR       0.0000 CMDDT     0.0000 CMDSP     0.0000
CMDSB      0.0200 CMDG     -0.0200 CMDF      0.0300
CMGE       0.0000
```

Figure 7
Longitudinal Moments Menu

The user can enter any values for these coefficients, and store them for re-use at any time. Some example of model implementation times might be of interest; we have input a new, and untested, Basic model developed from wind tunnel data in less than ten minutes; it took us another ten minutes or so to find out that the data had a decimal point in the wrong place in several coefficients, but we had the final model flying in less than 30 minutes.

An Engineering model menu data table is shown in Figure 8.

```
FUNCTION CLFO 5 X 3
ARGUMENTS
VARBPT     MACH
VARBPT     ALT3

           CLFO      MACH      ALT3

 1        0.3814    0.1000    0.0000
 2        0.3814    0.2000
 3        0.2200    0.4000
 4        0.1000    0.6000
 5        0.0800    0.8000
 6        0.3814    0.1000    15000.0000
 7        0.3814    0.2000
 8        0.4500    0.4000
 9        0.2000    0.6000
10        0.1300    0.8000
11        0.3814    0.1000    35000.0000
12        0.3814    0.2000
13        1.1000    0.4000
14        0.5000    0.6000
15        0.4000    0.8000
```

Figure 8
Engineering Model Data Table

In this menu the function CLFO has been defined to have two dimensions; these are called arguments in our system. As can be seen, the breakpoint values are simply entered as a numeric table. There are various automatic break point creation utilities, and the number of break points is completely flexible. A complex model can be entered without any specialized skills, obviously aero engineering capabilities must be used to validate the model and correct any data entry mistakes, but the actual input can be safely left to any person comfortable with text editors. It can take a week or so to enter a complex model by hand due to the amount of data involved, but this should be compared to the six months or more that are usually required. Savings are also apparent in the modification of existing models; all the requirements for specialized software skills are gone, and even a complex model can have coefficient changed in a matter of seconds.

In order to allow full use of our system, we developed a number of peripheral utilities. The most useful of these is the flight data recorder that emulates the operation of a typical digital data system. The recorder is operated under pilot command, and allows the selection of data elements and sampling rate for storage. This can process the data for data reduction purposes, and provides a strip chart emulator.

Conclusion

The development of the data driven structure was a by-product of the distributed micro-processor architecture that we created, and it has been very well received by a large cross section of users. Academies and test pilot schools find the system invaluable in the demonstration of aerodynamic concepts, and often use it to replace or supplement flight time in different aircraft. It has been used to allow students to conceptually build new aircraft as part of aerodynamic engineering courses, and prepare students for variable stability aircraft demonstrations and flights. It is invaluable in the Test and Evaluation environment, where test conditions can be pre-flown with minimum impact on flight schedules, and test point scheduling optimized to save valuable flight time. The ability to have a readily accessible simulator can, of course, only enhance flight safety benefits.

In conclusion, we feel that the development of this type of model structure has been a step towards reducing some of the complexity, expense and time associated with real time mathematic models. The continual re-writing of the equations of motion for each new model developed, and the difficulty that faces the aero engineers in accessing the coefficient data are unnecessary burdens. Simple to use does not necessarily mean that the system cannot also have excellent real time performance; in the final analysis, the ability of the flight test or aero engineer to develop models quickly and easily can only help the overall use of simulation facilities.

89-3310-CP

AN ALTERNATE APPROACH TO TABLE LOOK-UP ROUTINES FOR REAL-TIME DIGITAL FLIGHT SIMULATION

Michael K. Sinnett[1], **James E. Steck**[2],

Bruce P. Selberg[3], and **Robert B. Oetting**[3]

Department of Mechanical and Aerospace
Engineering, and Engineering Mechanics
University of Missouri-Rolla
Rolla, MO 65401

ABSTRACT

One computationally intense aspect of real-time digital flight simulation is the table look-up process for the build-up of aerodynamic coefficients and stability derivatives. This process consists of two parts; the table search routine, and the numerical interpolation routine used to approximate the needed value. These problems have typically been solved by employing top-down searches and index-ratio linear interpolation schemes, both of which are computationally intense. An alternate table search method, based on tracking the velocity of the lookup point through the data table, is presented here and compared to current search methods. Two new interpolation procedures are also presented and compared to currently used interpolation methods. These procedures are based on the approximation of the needed value over a subregion of the table by a polynomial which is linear in each of the independent variables. These methods yield the same numerical results as the currently used linear interpolation methods, but with fewer floating point calculations. It is shown that the judicious use of these new methods can lead to a significant decrease in table look-up frame time.

INTRODUCTION

The increase in complexity of flight simulators in the last decade has placed an increasingly larger computational burden on the simulation host computer. In the past, this has been dealt with by using separate processors for different tasks. This is not always financially feasible, especially in a university setting, where an entire simulation might be limited to one or two processors. While a perturbation model, sometimes used in a part task trainer, can be regarded as a linear system, the total force and moment model of a man-in-the-loop simulation requires a more complex mathematical representation of the vehicle. Coefficients and derivatives are often functions of several variables, including control surface position, angle of attack, Mach number, and several other parameters. These functions can seldom be represented in closed form by a reasonably small number of equations. Usually a table is created which consists of an array of dependent variables as a function of one or more independent variables. Table 1 is an example of such a function table. On each pass through the simulation program, a search is initiated through the appropriate tables, and an interpolation between known values is performed in each of the independent variable directions. This has been shown to be one of the most computationally intense aspects of the simulation process.[1] In order to retain model complexity and still achieve an acceptable frame rate, table look-up algorithms which reduce the computational load are highly desirable.

Two widely used search procedures are outlined by Rolfe and Staples.[2] The first performs a top-down search between successive array values, and exits once the proper interval is located. The second procedure also performs a top-down search, but only if the position in the table has changed significantly. This is a method which has been used in industry.[3] The search procedure is employed for each independent variable in a multi-dimensional table, and the process is repeated for each simulation pass.

[1] NSF Fellow, Member AIAA

[2] Member AIAA

[3] Professor, AIAA Associate Fellow

Table 1 A Two-Dimensional Table Example

Flap Position (Deg)	Sectional Lift Coefficient				
	Angle of Attack (Deg)				
	0	2	4	6	8
0.0	0.4	0.6	0.8	1.0	1.2
5.0	0.9	1.2	1.5	1.7	1.9

GAW-1 Airfoil with a 30% Chord Single Slotted Fowler Flap. Taken from [2].

Once the proper location in the array is reached, the dependent variable must be calculated. This dependent variable is considered to be a function, f, of n independent variables. It is calculated as outlined in [2] and [3] by carrying out successive linear interpolations in each of the n directions.

Alternate algorithms presented in this paper reduce the table search time by performing a top-down search during only the initialization pass. On all subsequent passes, the algorithm returns to the interval where the value will most likely reside. This is accomplished by tracking the rate of change of the magnitude and the direction of each independent variable as the dependent variable moves through the table. This is referred to as tracking the velocity of the variable. Once in the correct interval, the alternate methods presented here approximate the function, f, over a given subregion of the table, by a polynomial which is linear in each independent variable.

All of the above methods produce the same numerical results, but the algorithms presented here require fewer floating point calculations during the real-time portion of the simulation. When used in appropriate situations, these algorithms can significantly reduce look-up frame times.

ALGORITHM DESCRIPTIONS

Four separate algorithms were developed to investigate table look-up computation speeds and data storage requirements. The first two, Methods 1a and 1b, are based on methods presented by Rolfe and Staples.[2] The table look-up process consists of two distinct phases. The first is the table search, which is used to find the proper location in an n-dimensional table. Method 1a performs a top-down search between successive array values and exits once the proper interval is located. This search is performed for each independent variable, n search processes in an n-dimensional table look-up. The entire process is repeated on each simulation pass. Method 1b performs the top-down search procedure only if the interval in which the independent variable resides has changed since the previous pass.

The alternate methods presented here, Methods 2a and 2b, execute the top-down search only during the initialization pass. From then on, Methods 2a and 2b return to the same table interval as on the previous pass, where the independent variable will still most likely reside. If the interval in which one of the independent variables was located has changed in one of the n dimensions, the new algorithms in Methods 2a and 2b calculate the velocity of that independent variable through the table, and proceed directly to the correct new interval. Little testing or looping is involved, except in cases of a discontinuity in the aircraft flight condition due to impulsive loading or a discrete cockpit event.

The second phase of the look-up process is the calculation of the dependent variable, which is considered to be a function, f, of n independent variables, X_1, X_2, ... X_n. The function is represented by Method 1a and 1b as a data table, an array of values of f at different locations of the independent variables, spaced at intervals throughout the domain in which f is defined. The domain of f is divided into a number of 'rectangular' subregions, each corner of which is a point at which the function is defined. Figure (1) is an example of a two-dimensional subregion from Table 1.

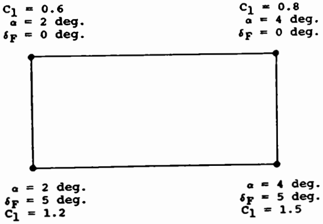


Figure 1. A Typical Two Dimensional Table Subregion.

When a value for f is needed by one of the simulation routines, the table search locates the proper rectangular subregion, and in Methods 1a and 1b, a linear interpolation scheme is employed to approximate f in terms of stored values at points nearby. Successive interpolations must be carried out for each of the independent variables, one for X_n, two for X_{n-1}, up to 2^{n-1} interpolations for X_1. This procedure is illustrated in Figure 2.

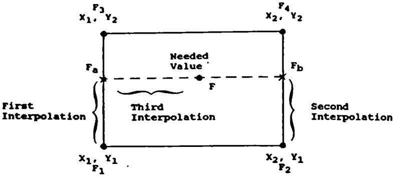


Figure 2. A Two-Dimensional Linear Interpolation Procedure.

Methods 2a and 2b recognize that the effect of the successive interpolations in Methods 1a and 1b is to approximate the function f over the given subregion by a polynomial which is linear in each independent variable X_i. It is further recognized that for an n-dimensional table, each subregion has 2^n corners, at which the value of f and the values of the independent variables X_1, X_2, ... X_n are known. The required n-linear polynomial has 2^n coefficients, one for each term. Thus for each subregion in an n-dimensional table, we have 2^n equations in 2^n unknowns, and the coefficients are uniquely determined. A polynomial expression to approximate the function, f, in the given subregion is then known. For example, Figure (1) would yield a system of four equations of the form

$$C_1 + C_2\alpha + C_3\delta_F + C_4\alpha\delta_F = C_l \qquad (1)$$

to be solved for the four unknowns C_1, C_2, C_3, C_4. This yields the bilinear polynomial expression for C_l in the given subregion

$$C_l = 0.40 + 0.10\alpha + 0.10\delta_F + 0.01\alpha\delta_F. \qquad (2)$$

This equation requires only 7 floating point operations to calculate a value for C_l, as compared to the 15 required for the linear interpolation scheme of Methods 1a and 1b for a 2-D table look-up.

Method 2a precalculates the polynomial coefficients for all the subregions in the table prior to real-time simulation. This can be done during an initialization pass, or the tables can be preprocessed, the coefficients stored, and then read as input for the actual simulation. One drawback to Method 2a is the data storage required for the polynomial coefficients of each table. For a two dimensional table, four coefficients are needed for each subregion. An interpolation table of n by m values requires 4(n-1)(m-1) coefficients to be stored for Method 2a. A four dimensional table would require 16 coefficients for each subregion, or 16(k-1)(m-1)(n-1)(p-1) coefficients for a table of k by m by n by p values. It should be emphasized that search speed is not affected by these larger arrays, since searching is performed on only the independent variable arrays.

Method 2b uses the same table search procedure and polynomial representation of the interpolation as Method 2a, but it does not precalculate the polynomial coefficients prior to real-time simulation. Instead, each time an interpolation point enters a new table subregion, the coefficients for only that subregion are calculated and stored. This virtually eliminates the additional storage required for the polynomial coefficients. For example, a four dimensional look-up would require the storage of only 16 coefficients at any one time. The obvious disadvantage of this method is the additional computation time required for the calculation of coefficients when entering a new subregion.

COMPUTATIONAL RESULTS

The four algorithms studied were coded in FORTRAN for 2-D, 3-D, and 4-D table lookups, and test cases were run on an Apollo DN10000. The 2-D tables were 14 x 12 in size, the 3-D tables were 14 x 12 x 12 in size, and the 4-D tables were 14 x 12 x 12 x 12. Over 3 million calls to the look-up routines were made in each set of test cases, and the frame speed was determined by dividing the total execution time by the total number of calls. The test cases differed in how frequently the dependent variable moved from one independent variable subregion into another. In situations where a new subregion was entered during only one percent of the total number of look-up calls, the look-up path employed is described as a low speed pass through the tables. A moderate speed pass describes cases where a new subregion is entered during 5 percent of the total number of look-up calls, and a high speed pass refers to a case where a new subregion is entered during 10 percent of the calls.

The results of the low speed pass through the tables are presented in Table 2. For the 2-D case, Method 1a is the slowest.

Table 2 Low Speed Look-up Path

Method	Algorithm Frame Speed (μsec)		
	2-D	3-D	4-D
1a.	21.7	33.2	47.8
1b.	12.7	20.0	32.6
2a.	9.1	17.6	36.3
2b.	7.9	15.0	38.6

This is because a top-down search is employed on each pass. Method 1b is faster because the top-down search is conducted only upon entrance into a new subregion. Since a 2-D linear interpolation requires 15 floating point operations and the calculation of a bilinear polynomial requires only 7, Methods 2a and 2b are both faster than Methods 1a and 1b. In order to calculate the coefficients of the bilinear polynomial required for the 2-D case, Method 2b requires the solution of a system of 4 equations in 4 unknowns each time a new subregion is entered. These coefficients are stored in a one-dimensional array. Method 2a precalculates these coefficients, but they are stored in a three-dimensional array. For the low speed 2-D case, it happens that the overhead due to the calculation of the coefficients is lower than the overhead due to the integer operations associated with manipulating the large three-dimensional array of coefficients. Thus, for the 2-D low speed case, Method 2b is faster than Method 2a. These same comments apply also to the 3-D case, where the solution of a system of 9 equations in 9 unknowns requires less overhead than the integer manipulation of the large four-dimensional coefficient array. In the 4-D low speed case however, the overhead required to solve a system of 16 equations in 16 unknowns, and the integer overhead required to manipulate the five-dimensional array of coefficients, are both greater than the time required to perform the 15 successive linear interpolations necessary in a 4-D look-up, and Method 1b remains the fastest in this 4-D case.

The results of the medium speed pass through the tables are presented in Table 3. In this set of test cases, the dependent variable is moving into new dependent variable subregions more often, and the added computational overhead of the calculation of coefficients of Method 2b becomes greater than the overhead of the integer manipulations of Method 2a. This gives Method 2a the edge in speed in both the 2-D and 3-D cases. Again, in 4-D situations, Method 1b remains the fastest, for the same reasons cited in the first set of test cases.

Table 3 Moderate Speed Look-up Path

Method	Algorithm Frame Speed (μsec)		
	2-D	3-D	4-D
1a.	21.7	33.2	47.6
1b.	12.9	20.3	32.7
2a.	9.3	17.8	36.3
2b.	10.5	27.0	112.1

The results of the high speed pass are presented in Table 4, with the same trends present as in Table 3.

Table 4 High Speed Look-up Path

Method	Algorithm Frame Speed (μsec)		
	2-D	3-D	4-D
1a.	21.7	33.3	47.6
1b.	13.1	20.5	32.9
2a.	9.4	18.0	36.3
2b.	13.8	42.0	203.3

Because of the increase in the number of subregion changes, Method 2b is slower than both Methods 1b and 2a for the 2-D case. Method 2a is the fastest for the 2-D and 3-D cases, and Method 1b retains the computational edge for 4-D table look-ups.

The use of Method 2a increases the data storage requirements for reasons outlined in the previous section. For the two-dimensional 14 x 12 table, the required storage increases from 0.7K to 2.3K when changing from Method 1b to Method 2a. The data storage requirement increases from 8K to 50K for a three-dimensional 14 x 12 x 12 table, and from 97K to 1100K for a 14 x 12 x 12 x 12 table. Method 2b requires virtually no additional storage.

RECOMMENDATIONS AND CONCLUSIONS

The use of Method 2a is recommendèd in 2-D look-up situations where the look-up point is moving rapidly through the tables. It is the fastest method in these situations, and the increase in data storage for the coefficients is minimal. If the look-up point is moving slowly through the 2-D tables, the use of Method 2b is recommended, as it is the fastest method in this situation and carries almost no additional storage requirements for the polynomial coefficients.

The use of Method 2a is also recommended in 3-D look-up situations where the look-up point is moving quickly through the tables, but the amount of available memory could be a concern. While this is the fastest method available for these situations, the storage requirement for Method 2a increases substantially over Method 1b. Method 2b can alleviate this problem, but its use should be limited to circumstances where the look-up point is moving into new intervals in only one or two percent of the total number of look-up passes.

Method 2a is not recommended for use in any 4-D (or higher) look-up situations, because it is slower than the other methods. Also, changing from Method 1b to Method 2a for a 4-D table drastically increases the storage requirements. Hence, Method 2a could only be a viable option in a 4-D look-up situation if data storage is not a problem and if the simulator host computer has extremely high integer performance. Method 2b is also not recommended in any circumstance for a 4-D (or higher) look-up because frame speeds are too high. Method 1b remains the most suitable for a 4-D table look-up.

The development of these new algorithms supplies the simulation engineer with another set of tools fòr table look-up applications. Like any tools, they must be used carefully, and in only the appropriate situations. For example, the use of Method 2b would be inappropriate in situations where the look-up point is moving rapidly through the data tables. The calculation of rolling moment due to aileron deflection is such a situation, as rolling moment is a function of roll rate, a quantity which can change rapidly. All of the methods described produce the same numerical results, but the new algorithms presented here require fewer floating point calculations during certain real-time portions of the simulation. When used in appropriate situations, these algorithms can significantly reduce look-up frame times.

REFERENCES

1. Forsstrom K. S., "Array Processors in Real-Time Flight Simulation", IEEE Paper 83-0062, 1983.

2. Rolfe J. M. and Staples K. J., Flight Simulation, Cambridge University Press, 1986.

3. McDonnell Aircraft Company, "Hybrid Library Summary", Flight Simulation Laboratory D-251, December 1982.

4. Wentz W. H. Jr. and Seetharam H. C., "Development of a Fowler Flap System for a High Performance General Aviation Airfoil.", NASA CR-2443, December 1974.

89-3311-CP

AN AIR COMBAT SIMULATION MODEL SUITABLE FOR THE EVALUATION OF AGILITY AND EFM

Carlos T. Miralles, John Selmon, Steven M. Trujillo

Rockwell International, Los Angeles, California

Abstract

This paper introduces a new digital air battle simulation model suitable for the evaluation of agility and enhanced fighter maneuverability (EFM). This six degree-of-freedom (6DOF) model includes the methodology to digitally control flight path and attitude while retaining the implicit flying qualities necessary for real-time man-in-loop (MIL) implementation. Designed for use in low cost piloted workstations and integration with the Advanced Air-to-Air System Performance Evaluation Models, this model represents a substantial improvement in design and system effectiveness analysis capability.

1. Introduction

Advances in aircraft design, weapons systems, avionics and ECM are dramatically affecting the operational utilization of fighter aircraft. Recent work has shown that close-in-combat (CIC) may be an unavoidable occurrence of future air combat[1]. As separation distance decreases, the time a pilot spends changing maneuver state increases (Figure 1). Agility, a measure of an aircraft's ability to change maneuver state, has become an important design consideration.

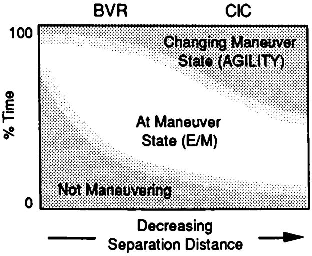


Fig. 1 Agility is an Important Design Consideration for CIC

Recent interest in the role of agility and Enhanced Fighter Maneuverability (EFM) on fighter combat effectiveness has led to various efforts throughout the industry and government to evaluate these technologies. Efforts range from studies to define agility and develop metrics to a full scale aircraft demonstration program with the Rockwell/MBB X-31. Qualitatively, all the proposed metrics/definitions reflect an aircraft's ability to change maneuver state in two or three principal axes[2]. Until now, there has not been a digital air battle simulation of sufficient fidelity to quantify the operational significance of agility and EFM.[3] Figure 2 is a qualitative illustration of the MBB definition of agility.

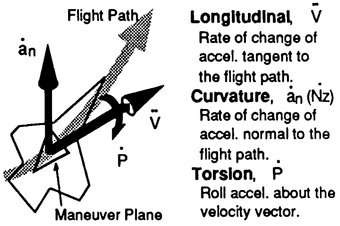

Flight Path
$\dot{a}_n$
$\dot{V}$
$\dot{P}$
Maneuver Plane
Longitudinal, $\dot{V}$
Rate of change of accel. tangent to the flight path.
Curvature, $\dot{a}_n$ (Nz)
Rate of change of accel. normal to the flight path.
Torsion, $\dot{P}$
Roll accel. about the velocity vector.

Fig. 2 Physical Representation of Agility

Tactical simulation models (i.e. AASPEM[4], Tac-Brawler, and others) were originally intended for the evaluation of beyond-visual-range (BVR) tactics and vehicle dynamics. The models use point-mass, 3DOF equations of motion and were driven by aerodynamic and propulsive performance variables reflecting energy-maneuverability (E/M) design parameters[4]. Use of the higher fidelity models necessary for MIL, agility and EFM studies in digital air battle simulations have been prohibitive due to the requirements for a flight control system, the increased computational requirements, and difficulties in implementing pointing or flight path control logic. In practice, CIC has been studied primarily through the use of piloted simulations in domes or stations with limited visuals. The repetitive process necessary for obtaining data of statistical significance using these tools proves very costly.

The need for a higher fidelity digital air battle simulation model consistent with the requirements for operational analysis and reflecting design parameters relevant to agility is evident[3]. Additionally, MIL capability will be required to build a knowledge data base of maneuvers and tactics from which digital air battle logic may be extracted. Graphically, the evolution of digital air battle simulation model fidelity is illustrated in Figure 3.

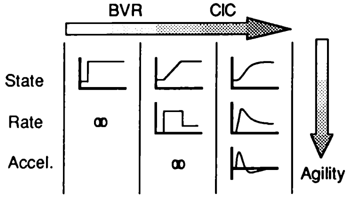


Fig. 3 Evolution of Digital Air Battle Simulation Model Dynamics

With the higher fidelity data base and more representative flight dynamics, additional logic is required to avoid unwanted departures and to maneuver the aircraft in a versatile, yet precise manner. A simple mechanism for digitally controlling the flight path and orientation in space is presented in this paper which may be used to define and control highly dynamic maneuvers in response to a dynamic environment.

As part of an ongoing effort at Rockwell International to evaluate agility and EFM, a high fidelity yet simple air vehicle simulation model was developed. The model is intended for use in:

- Operations analysis digital and/or MIL m vs. n air battle simulations.
- Digital flight path and maneuver optimization.
- MIL or digital tool for the development of tactics and displays to exploit new technologies.
- Design and analysis of future air vehicles.

This simulation tool allows for a higher fidelity model with implicit conventional (or user defined) flying qualities without the need for a flight control system or the computational requirements of typical high order simulations.

2. Nomenclature

Symbols:

α	Angle-of-attack, (deg)
β	Angle-of-sideslip, (deg)
ϕ	Body axis bank angle, (deg)
θ	Body axis pitch attitude, (deg)
ψ	Body axis heading angle, (deg)
μ	Stability axis bank angle, (deg)
γ	Glide slope angle, (deg)
χ	Stability axis heading angle, (deg)
P	Roll rate, (deg/sec)
Q	Pitch rate, (deg/sec)
R	Yaw rate, (deg/sec)
M	Moment, (ft·lbs)
Ω	Angular rate, (deg/sec)
I	Mass moment of inertia, (slug·sqft)
m	Mass, (slugs)
$\bar{m}$	Maneuver Plane Vector
g	Accel. due to gravity, (deg/sec^2)
n	Normal acceleration, (g's)
F	Force, (lbs)
L	Lift, (lbs)
T	Thrust, (lbs)
i_T	Thrust Incidence Angle, (deg)
V	Velocity, (ft/sec)
u,v,w	Velocity components
q_{dyn}	Dynamic pressure, (psf)
Sref	Reference wing area, (sqft)
c	Mean aerodynamic chord, (ft)
b	Reference wing span, (ft)
CL	Stability axis lift coefficient
CD	Stability axis drag coefficient
Cl	Body axis rolling moment coef.
Cm	Body axis pitching moment coef.
Cn	Body axis yawing moment coef.
D_{ew}	Earth-wind direction cosine matrix
d_{rc}	Direction cosine from D_{ew} where r=row and c=column
e	Quaternion, Exponent
r	Target vector
i, j, k	Unit vector notation
t	Time, (sec)
δ or Δ	Derivative or increment
ω	Frequency, (rad/sec)
ζ	Damping Ratio
τ_{rm}	Roll mode time constant, (sec)
Λ, Γ	Dummy variables

Subscripts:

e	Earth axis
b	Body axis
s	Stability axis
w	Wind axis
x, y, z	Axis labels
max	The maximum value
min	The minimum value
cmd	Commanded value
ss	Steady state value
TV	Thrust vectoring
sp	Short period
+	Positive value
-	Negative Value

Superscripts:

–	Vectors
.	Derivative w.r.t. time

3. General Description of Model

This simulation is designed to facilitate changes in model fidelity and in the dynamic characteristics of the aircraft response. In keeping with the operations analysis requirements, a minimum of information is saved from frame to frame even for high fidelity models.

A quaternion flight trajectory model is the building block at the core of the simulation model. The 6DOF model is driven by the wind-axis forces and roll rate about the velocity vector. These forces and the roll rate reflect the aircraft's aerodynamic performance, maneuverability, and propulsion system capability.

The simplest utilization of this simulation uses the direct control of the quaternion forces and the roll rate as determined by the the capabilities defined in the data base. As such, this implementation does not differ significantly from existing air battle simulation models when the forces are constrained by a limit on their rate of change.

Other applications may require the use of a higher fidelity data base and/or classical flying qualities (i.e. short period, roll mode, dutch roll mode or user defined dynamics) in the aircraft model. To accomplish this, a controller/integrator/predictor (CIP) was devised to assure captures of commanded angle-of-attack, angle-of-sideslip or change in bank angle without a complicated flight control system. The CIP calculates average forces and rates at each time interval for use in the flight trajectory calculations. Using the CiP in simple applications as well as for high fidelity models reduces the dependency on frame rate for the flight trajectory.

A summary block diagram of the principal operations is illustrated in Figure 4.

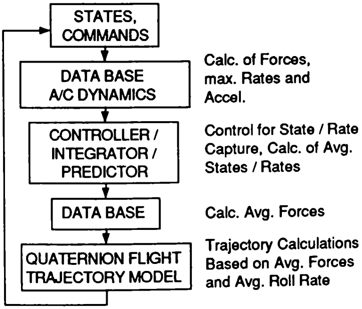

STATES, COMMANDS
DATA BASE A/C DYNAMICS
Calc. of Forces, max. Rates and Accel.
CONTROLLER / INTEGRATOR / PREDICTOR
Control for State / Rate Capture, Calc. of Avg. States / Rates
DATA BASE
Calc. Avg. Forces
QUATERNION FLIGHT TRAJECTORY MODEL
Trajectory Calculations Based on Avg. Forces and Avg. Roll Rate

Fig. 4 Block Diagram

A description of the quaternion flight trajectory model, CIP, and and example of a high fidelity data base with implicit flying qualities are presented in the following sections. The example includes the effects of aerodynamic control power, thrust vectoring and inertial coupling. Additionally, classical dynamics in the short period and roll modes are modelled for future modification to MIL by directly controlling the maximum roll rate. Although not all applications warrant use of a model of this complexity, this example illustrates the flexibility and adaptability of the method.

For the purpose of this paper, only the pitch and roll axis high fidelity models are described due to the abundance of simple propulsion models currently used. At the time of writing this paper, the yaw axis control and dutch roll dynamics have not been enabled or validated and $\beta=0$ degrees is assumed. Future studies on the effectiveness of nose-pointing, post stall technologies and tactics, and the all-aspect missile will utilize control of all 6DOF.

4. Quaternion Flight Trajectory Model

Use of a quaternion solution for the modeling of aircraft trajectories in the simulation avoids the discontinuities and ambiguities associated with traditional Euler angle solutions. This is accomplished by considering the transformation from one reference frame to another, in this case from earth to wind axis systems, as a single rotation about some axis. In the quaternion e, the rotation angle is indirectly represented by the first scalar element e_0 and the axis of rotation by the vector [e_1, e_2, e_3].

The dynamics of the quaternion as implemented in the model are governed by the differential equation:

$$\frac{d}{dt}\begin{pmatrix} e_0 \\ e_1 \\ e_2 \\ e_3 \end{pmatrix} = \frac{1}{2}\begin{pmatrix} -e_1 & -e_2 & -e_3 \\ e_0 & -e_3 & e_2 \\ e_3 & e_0 & -e_1 \\ -e_2 & e_1 & e_0 \end{pmatrix}\begin{pmatrix} P_w \\ Q_w \\ R_w \end{pmatrix} \qquad (1)$$

where P_w, Q_w and R_w are the wind axis roll, pitch and yaw rates, respectively. While P_w is a control in the model, Q_w and R_w are calculated from the total wind axis forces in the normal plane of the aircraft as follows.

The acceleration of the aircraft in the plane normal to the velocity vector is:

$$\bar{a}_{np} = \left(\frac{F_{yw}}{m} + g_{yw}\right)\bar{j}_w + \left(g_{zw} - \frac{F_{zw}}{m}\right)\bar{k}_w \qquad (2)$$

where g_{yw} and g_{zw} are the y- and z-components of gravity in the wind axis system, respectively, and F_{yw} and F_{zw} are the total aerodynamic and thrust forces acting on the aircraft in those directions. The angular velocity of the normal plane due to this acceleration is calculated by the cross product:

$$\bar{\Omega}_{np} = \frac{\bar{i}_w \times \bar{a}_{np}}{V} \qquad (3)$$

or, operationally:

$$\bar{\Omega}_{np} = \det \begin{bmatrix} \bar{i}_w & \bar{j}_w & \bar{k}_w \\ \frac{1}{V} & 0 & 0 \\ 0 & \left(\frac{F_{yw}}{m} + g_{y\,w}\right) & \left(g_{zw} - \frac{F_{zw}}{m}\right) \end{bmatrix} \qquad (4)$$

Expansion yields:

$$\bar{\Omega}_{np} = \frac{1}{V}\left(\frac{F_{zw}}{m} - g_{z\,w}\right)\bar{j}_w + \frac{1}{V}\left(\frac{F_{yw}}{m} + g_{y\,w}\right)\bar{k}_w \qquad (5)$$

It therefore follows that:

$$\bar{Q}_w = \frac{1}{V}\left(\frac{F_{zw}}{m} - g_{z\,w}\right)\bar{j}_w \qquad (6)$$

and:

$$\bar{R}_w = \frac{1}{V}\left(\frac{F_{yw}}{m} + g_{y\,w}\right)\bar{k}_w \qquad (7)$$

Thus, by knowing the forces acting on the aircraft and the wind axis roll rate, the above differential equation can be evaluated and the quaternions updated at each time step.

From the quaternions, it is possible to evaluate the direction cosine matrix D_{ew} relating the wind and earth axis systems such that:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix}_w = D_{ew}\begin{pmatrix} x \\ y \\ z \end{pmatrix}_e \qquad (8)$$

It is well-known that:

$$D_{ew} = \begin{pmatrix} \cos\gamma\cos\chi & \cos\gamma\sin\chi & -\sin\gamma \\ \sin\mu\sin\gamma\cos\chi - \cos\mu\sin\chi & \sin\mu\sin\gamma\sin\chi + \cos\mu\cos\chi & \sin\mu\cos\gamma \\ \cos\mu\sin\gamma\cos\chi + \sin\mu\sin\chi & \cos\mu\sin\gamma\sin\chi - \sin\mu\cos\chi & \cos\mu\cos\gamma \end{pmatrix} \qquad (9)$$

Further, it can be shown[5] that:

$$D_{ew} = \begin{pmatrix} e_0^2+e_1^2-e_2^2-e_3^2 & 2(e_1e_2+e_0e_3) & 2(e_1e_3-e_0e_2) \\ 2(e_1e_2-e_0e_3) & e_0^2-e_1^2+e_2^2-e_3^2 & 2(e_0e_1+e_2e_3) \\ 2(e_1e_3+e_0e_2) & 2(e_2e_3-e_0e_1) & e_0^2-e_1^2-e_2^2+e_3^2 \end{pmatrix} \qquad (10)$$

It is thus simple to determine the wind axis pitch and heading angles γ and χ:

$$\gamma = \sin^{-1}\left[\,-2(e_1e_3 - e_0e_2)\,\right] \qquad (11)$$

$$\chi = \tan^{-1}\left[\frac{2\,(e_1e_2 + e_0e_3)}{e_0^2 + e_1^2 - e_2^2 - e_3^2}\right] \qquad (12)$$

The wind axis bank angle μ is determined in a similar manner, but with an additional step imbedded to guarantee proper sign:

$$\mu = \sin^{-1}\left[\frac{2(e_0e_1+e_2e_3)}{\sqrt{(e_0^2+e_1^2-e_2^2-e_3^2)^2 + 4(e_1e_2+e_0e_3)^2}}\right] \qquad (13)$$

Velocity is updated by evaluating the differential equation:

$$\dot{V} = \frac{F_{xw}}{m} + g_{xw} \qquad (14)$$

where g_{xw} is the x-component of gravity in the wind axis and F_{xw} is the total aerodynamic and thrust force in the $\bar{i}_w$ direction. Position is similarly updated via the expressions:

$$\dot{x}_e = V_{xe} \qquad (15)$$

$$\dot{y}_e = V_{ye} \qquad (16)$$

$$\dot{z}_e = V_{ze} \qquad (17)$$

where V_{xe}, V_{ye} and V_{ze} are the x- y- and z-components of velocity in the earth axis system.

With minimal input - total forces and wind axis roll rate - the quaternion model produces an accurate and concise representation of aircraft trajectory.

5. Controller

The Controller/Integrator/Predictor (CIP) is principally responsible for insuring a capture of the commanded α, β or change in bank angle constrained by the maximum usable rates and accelerations defined by the dynamics model and/or data base. The operation predicts the dynamics to the completion of the capture using starting (current) and target (commanded) conditions to analytically determine a solution. Figure 5 illustrates the process and labels the parameters defined by the dynamic response model.

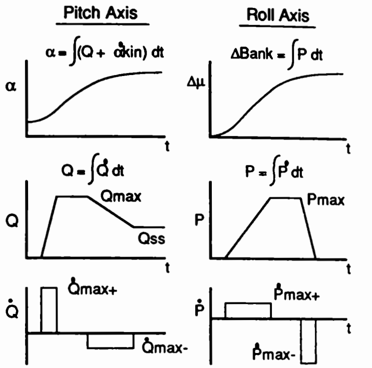


Fig. 5 A Notional Illustration of the CIP

Once a solution has been obtained, the functions are evaluated over the current time step and the average roll rate, angle-of-attack and sideslip are calculated. As the frame rates typical of digital air battle simulations are very low (0.1 sec, compared with 0.01 to 0.03 for other applications), the averaged values tend to produce kinematics less sensitive to frame rate and more representative of the higher order dynamics used to resolve the capture between the time slices. Figure 6, showing a comparison between time histories run at δt=0.10 and 0.05, is an example of the insensitivity.

An important feature incorporated into the model is the representation of the terms which account for the effects of n_z, gravity and thrust in the transformation from the rate equations in the dynamics model to the aerodynamic angles (indicated by the variable $\dot\alpha_{kin}$ in Figure 5 and $\dot\beta_{kin}$ for the yaw axis). The effect of these terms is to account for the additional control power required to control α and β at very low airspeeds and unusual attitudes. As implemented, the complete $\dot\alpha$ expression is:

$$\dot\alpha = \dot\alpha_{TH} + \dot\alpha_{GR} - \dot\alpha_{nz} - P_s\tan\beta + Q_s \qquad (18)$$

where:

$$\dot\alpha_{TH} = \frac{1}{V\cos\beta}\frac{T}{m}(\sin i_T \cos\alpha - \cos i_T \sin\alpha) \qquad (19)$$

$$\dot\alpha_{GR} = \frac{1}{V\cos\beta}\, g\,(\cos\Theta \cos\Phi \cos\alpha + \sin\Theta \sin\alpha) \qquad (20)$$

$$\dot\alpha_{nz} = \frac{1}{V\cos\beta}\frac{S_{ref}}{m}\, q_{dyn}\, CL \qquad (21)$$

The expression for $\dot\beta$ is similar in form. To illustrate the effect of these terms, Fig. 7 shows an aircraft at high altitude and low airspeed, with no thrust and a commanded α = 20 degrees.

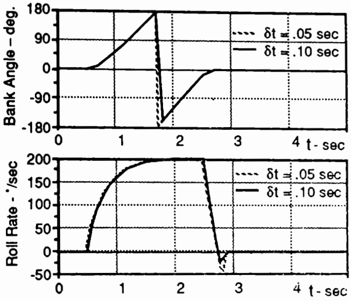


Fig. 6 The Model is Insensitive to Frame Rate

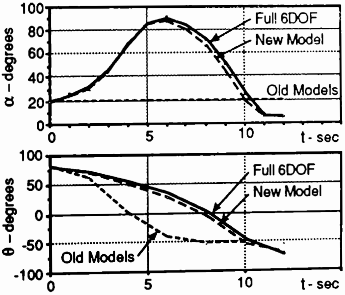


Fig. 7 Departure Prediction

Without thrust vectoring or aerodynamic control power (low q_{dyn}) the aircraft would be expected to depart even while maximum nose down control was applied in an effort to maintain the commanded 20 degrees α. The previous AASPEM model did not predict a departure as the kinematic model is decoupled from the α control. A comparison of a high fidelity full 6DOF flight simulation with the new model show much better agreement. It is clear that while evaluating post-stall-technologies (PST) or developing design criteria in the far left portion of the flight envelope, this effect would greatly alter the nose-pointing capability of a configuration with marginal control or agility. As might be expected, departures would occur any time the maximum available acceleration (control power) becomes insufficient.

6. Dynamic Response Model

The short period and roll modes of an aircraft may be approximated by use of linear first and second order differential equations. As the goal of a flight controls designer is to achieve the response simplicity of these low order models, we may use the models to represent idealized aircraft dynamics.

For this example, the idealized model characteristics will be compared with the available control power. Although this extension to the method is not essential to the operation of the simulation, the effects of thrust vectoring or other control power enhancements and inertial coupling on the operational utilization of PST are important design considerations.

Beginning with the roll axis, a classical first order roll mode response to a step input is used as an idealized model (Eqn. 22). The relationships between the maximum roll acceleration, τ_{rm} and the maximum roll rate is will be used to determine the desired roll acceleration for a given τ_{rm} and P_{max}.

$$P = P_{max} \;\; (1 - e^{(-t/\tau rm)}) \qquad (22)$$

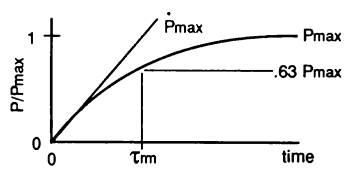


Fig. 8 First Order Roll Mode

The peak roll acceleration for the idealized response is found by differentiating Eqn. (18) and evaluating at t=0.

$$\dot{P}_{max} = P_{max} / \tau_{rm} \qquad (23)$$

Accounting for the acceleration due to damping, Eqns. (24-25) may be used to calculate the maximum body axis roll acceleration at any time during a maneuver.

$$\dot{P}_{max+} = \frac{1}{\tau_{rm}} (P_{max} - P) \qquad (24)$$

$$\dot{P}_{max-} = \frac{1}{\tau_{rm}} (- P_{max} - P) \qquad (25)$$

For a digitally controlled simulation, Pmax would be considered the maximum roll rate as defined in the data base. And for the MIL application, Pmax is equivalent to the commanded roll rate by the pilot.

To include the effects of control power and inertial coupling, we will consider the moment equations in the body axis:

$$\dot{P} = M_x / I_x \qquad (26)$$

$$\dot{Q} = M_y / I_y \qquad (27)$$

$$\dot{R} = M_z / I_z \qquad (28)$$

In this example, the effects of thrust vectoring and aerodynamic control power on agility and PST are of interest. Expanding Eqns. (26) through (28) to include the relevant terms:

$$\dot{P} = \frac{(I_y - I_z)}{57.3\, I_x} Q R + \frac{57.3\, S_{ref}\, b}{I_x} q_{dyn}\, Cl_{tot} + \frac{S_{ref}\, b^2}{2\, I_x} \frac{q_{dyn}}{V} Cl_p\, P \qquad (29)$$

$$\dot{Q} = \frac{(I_z - I_x)}{57.3\, I_y} P R + \frac{57.3\, S_{ref}\, c}{I_y} q_{dyn}\, Cm_{tot} + \frac{S_{ref}\, c^2}{2\, I_y} \frac{q_{dyn}}{V} (Cm_q\, Q + Cm_{\dot{\alpha}} \dot{\alpha}) \qquad (30)$$

$$\dot{R} = \frac{(I_x - I_y)}{57.3\, I_z} P Q + \frac{57.3\, S_{ref}\, b}{I_z} q_{dyn}\, Cn_{tot} + \frac{S_{ref}\, b^2}{2\, I_z} \frac{q_{dyn}}{V} Cn_r\, R \qquad (31)$$

The maximum control power available may be calculated by substituting the aerodynamic data for the maximum control deflection in each direction. The contribution due to thrust vectoring may be included in Cl_{tot}, Cm_{tot} and Cn_{tot}.

$$\dot{P}_{max+} = \dot{P} \text{ (max. right control power)} \qquad (32)$$

$$\dot{P}_{max-} = \dot{P} \text{ (max. left control power)} \qquad (33)$$

$$\dot{Q}_{max+} = \dot{Q} \text{ (max. nose-up control power)} \qquad (34)$$

$$\dot{Q}_{max-} = \dot{Q} \text{ (max. nose-down control power)} \qquad (35)$$

$$\dot{R}_{max+} = \dot{R} \text{ (max. right control power)} \qquad (36)$$

$$\dot{R}_{max-} = \dot{R} \text{ (max. left control power)} \qquad (37)$$

For this case β=0 ($\dot{\beta}$=0). The expression defining the conditions for a coordinated roll about the velocity vector is:

$$R = P \tan \alpha \qquad (38)$$

To satisfy this condition, a check on the maximum available control power (Eqns. (32-37)) is required. The minimum set defines the available coordinated roll acceleration. The relationship is graphically illustrated in Figure 9.

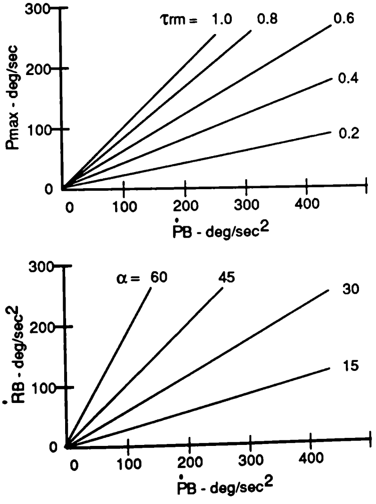


Fig. 9 The Relationship Between Body Axis Roll and Yaw Acceleration for a Coordinated Roll ($\dot{\alpha}$=0)

Similarly, the desired roll acceleration from Eqns. (24-25) may now be compared with the maximum available coordinated roll acceleration to obtain the usable roll acceleration. Transforming the body axis accelerations into the wind axis (β=0) for use by the CIP we get:

$$\dot{P}_{wmax+} = \dot{P}_{max+} \left(1 + \frac{\sin\alpha^2}{\cos\alpha} \right) \qquad (39)$$

$$\dot{P}_{wmax-} = \dot{P}_{max-} \left(1 + \frac{\sin\alpha^2}{\cos\alpha} \right) \qquad (40)$$

Taking a similar approach for the pitch axis, an idealized model response will be compared with the airframe capability to obtain the maximum usable pitch rate and accelerations. To simplify the idealized second order model and the data base requirements, a damping ratio of 0.85 is assumed (ζ=0.85, reasonably well damped). Also, since the CIP requires only the maximum usable rate and acceleration for each time step, an expression for the maximum pitch rate at any time was derived as a function of the pitch rate overshoot ratio (Q_{max}/Q_{ss}, supplied in the data base or derived from CAP) and the difference between the current and commanded states.

$$Qmax = \frac{57.3\,g}{V}\left(\left(\frac{Q_{max}}{Q_{ss}}\right) n_a(\alpha_{cmd} - \alpha) + n_\alpha \alpha - d_{33}\right) \qquad (41)$$

The maximum pitch acceleration may be related to the control anticipation parameter (CAP) by:

$$\dot{Q}_{max} = CAP\ (n_{zcmd} - n_z) \qquad (42)$$

or:

$$\dot{Q}_{max} = \frac{(57.3g/V)^2\, n_\alpha\, (n_{zcmd} - n_z)}{\left(\frac{1}{T\theta_2\, \omega_{sp}}\right)^2} \qquad (43)$$

where the denominator has been related to the maximum pitch rate overshoot ratio in Ref. 6.

Given flight test data, high fidelity simulation data or flying qualities design criteria, sufficient information should be available to define this function for the flight envelope.

The commanded (steady-state) pitch rate may now be determined using Eqn. (41) by setting the current states to the commanded values. The available pitch acceleration defined by Eqns. (34-35) may now be compared to the results from Eqn. (42 or 43). The minimum values represent the maximum usable pitch acceleration.

7. Maneuver Plane Control

To avoid the discontinuities inherent in commanding roll via the Euler roll angle μ, it was determined that a simple bank angle increment δ_{bank}, measured as a rotation about the velocity vector, would be used as the control. By using the relative position of the target, determined by tactics, and the direction of the velocity vector, a δ_{bank} can be calculated that is independent of the attitude of the aircraft.

Given a target defined by a target vector $\bar{r}$, it is easy to define a maneuver plane for the attacking aircraft by the unit vector $\bar{m}$, where:

$$\bar{m} = \frac{\bar{r} \times \hat{i}_w}{|\,\bar{r} \times \hat{i}_w\,|} \qquad (44)$$

where $\hat{i}_w$ is the unit vector parallel to the velocity of the attacking aircraft.

Considering the normal plane of the aircraft, the plane perpendicular to the velocity of the aircraft, the necessary δ_{bank} can be found. Here, $\bar{L}$ is the total external force acting on the aircraft in the $\bar{k}_w$ direction, and $\bar{g}_n$ is a vector with magnitude equal to and direction opposite the projection of gravity in the normal plane.

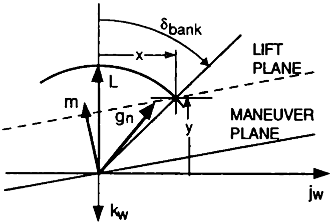


Fig. 10 The normal plane of the aircraft

The intersection of the maneuver plane with the normal plane is defined by the equation:

$$y = -\frac{m_{yw}}{m_{zw}}x \qquad (45)$$

where m_{yw} and m_{zw} are the magnitudes of the y- and z-components of $\bar{m}$ in the wind axis system. The lift plane's intersection with the normal plane is similarly defined by:

$$y = -\frac{m_{yw}}{m_{zw}}x - g{\cdot}d_{33} + g{\cdot}d_{23}\frac{m_{yw}}{m_{zw}} \qquad (46)$$

where d_{33} and d_{23} are elements (3,3) and (2,3) of the direction cosine matrix $\mathbf{D}_{ew}$, respectively. Note that in Eqns. (45) and (46), the generic term x is used to denote the direction parallel to $\hat{j}_w$ and y is parallel to $\hat{k}_w$.

Adopting the notation:

$$\Lambda = \frac{m_{yw}}{m_{zw}} \qquad (47)$$

and:

$$\Gamma = g\,(d_{23}\Lambda - d_{33}) \qquad (48)$$

the intersection of the arc swept out by $\bar{L}$ can be found by solving:

$$(\Lambda^2 + 1)x^2 - 2\Lambda\Gamma x + \Gamma^2 - L^2 = 0 \qquad (49)$$

where $L = |\,\bar{L}\,|$. Note that if $L > |\,\bar{g}_n\,|$, two real solutions exist. To determine which is correct, define $\bar{m}_\perp$ as $\bar{m}$ rotated +90° about $\bar{i}_w$ in the normal plane, that is,

$$\bar{m}_\perp = -\,m_{zw}\bar{j}_w + m_{yw}\bar{k}_w \qquad (50)$$

This is the direction in which $\bar{L}$ should turn the aircraft. Thus if $m_{zw} < 0$, the aircraft should turn right, and, likewise, if $m_{zw} > 0$, the aircraft should turn left.

If the solution for x in Eqn. (49) is greater than g_y, then the aircraft will bank right. Hence, if $m_{zw} < 0$, a right bank is desired and thus the solution for x should be chosen in which $x > g_y$; if $m_{zw} > 0$, the solution in which $x < g_y$ should be chosen.

Having chosen x as above, y can be calculated from Eqn. (46) which defines the lift plane. From these, it is easy to compute the required change in bank angle as:

$$\delta_{bank} = \frac{x}{|\,x\,|}\tan^{-1}\left(\frac{x}{-y}\right) \qquad (51)$$

with the arctangent function restricted to [0°, 180°].

8. Example Time Histories

Fig. 10 shows a comparison between flight test data[7] and the new simulation model. The aircraft is an F-16, at 1-g, Mach=0.58 and 10,100 feet altitude, performing a 360 degree roll with full stick deflection. The agreement is excellent as would be expected where the aircraft dynamics are well represented by a low order model.

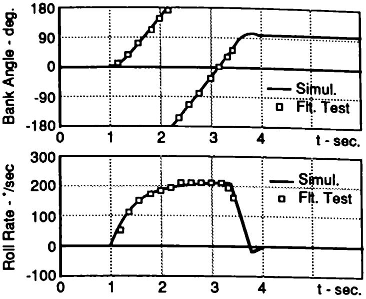
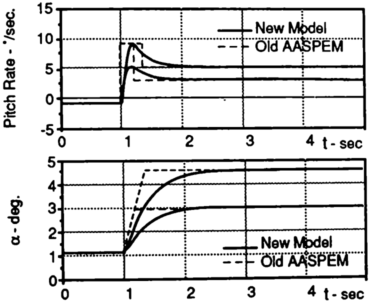


Fig. 10 Comparison of Flight Test and the New Model. F-16, Mach=0.58, Altitude=10,100 ft., 1-g 360 degree maximum performance roll.

A comparison of the pitch axis response to the previously used 3DOF models dramatically illustrates the inadequacy of the $\dot{\alpha}$ rate models to correlate an n_z capture time. The new model also demonstrates a more classical response proportional to the command lending itself well to MIL applications.



9. Summary

The integration of the quaternion flight trajectory model, CIP and high fidelity data base has provided the digital air battle simulation environment with a tool of sufficient fidelity to model agility and EFM, both MIL and digitally controlled, in a low cost environment.

A quaternion flight trajectory model has been defined which produces an accurate and concise representation of aircraft trajectory while avoiding the discontinuity and ambiguities associated with traditional Euler angle solutions. Dynamics representative of low order systems has been included as part of the data base to provide MIL capability and more representative flying qualities. In order to utilize the dynamics, a Controller/Integrator/Predictor was devised to assure a simultaneous capture of the commanded states and rates. The data required for this simulation model is adaptable to the situation. With as little as a table of maximum roll rates, roll mode time constants and pitch rate overshoot ratios, a realistic dynamic response suitable for MIL is achieved. Additional model fidelity to include the effects of thrust vectoring and inertial coupling may be incorporated through the moment equations.

Utilization of the maneuver plane as a control device has avoided the undesirable characteristics of traditional Euler angle commands. Additionally, this feature more easily accommodates future command sequencing driven by the piloting/tactics logic necessary for the highly dynamic CIC environment.

With this new simulation model, higher order dynamics may be represented, easily and quickly without a flight control system, while providing good flying qualities. This characteristic allows the model to be used for both digital and MIL tactics and maneuver development, which in turn, will provide more accurate and capable digital air battle simulations.

References

1. Herbst, W. B., "Dynamics of Air Combat," AIAA Journal, V. 20, 1983

2. Bitten, R., Rockwell International, "Qualitative and Quantitative Comparison of Government and Industry Agility Metrics," AIAA-89-3389

3. Hodgkinson, J. et al, "Relationships Between Flying Qualities, Transient Agility, and Operational Effectiveness of Fighter Aircraft," AIAA-88-4329-CP

4. McDonagh, G.M., "Advanced Air-to-Air System Performance Evaluation Model (AASPEM) Manual," Boeing Document D180-29122-1, November 1985.

5. Goodman, R. J., "Digital Simulation of Aircraft Flight Path Dynamics Using Quaternions," AIAA-78-311

6. DoD, MIL-STD- 1797 (USAF), "Flying Qualities of Piloted Vehicles," 31 March 1987.

7. Pape, J. K., and Garland, M. P., "F-16A/B Flying Qualities Full Scale Development Test and Evaluation," AFFTC-TR-79-10

89-3312-CP

DEVELOPMENT OF A TACTICAL GUIDANCE RESEARCH AND EVALUATION SYSTEM (TGRES)

Kenneth H. Goodrich[*] and John W. McManus[+]
NASA Langley Research Center, Hampton, Virginia

Abstract

The ongoing development of a tool for tactical guidance research and the analysis of airplane system performance in a tactically significant environment are described. The objective of the tool is to provide a means by which researchers can explore and exploit enhancements to high-performance airplane agility. The completed tool will include high-fidelity batch and piloted simulation capabilities, an advanced tactical guidance logic, and a user-friendly interface. While the tool is being developed for the purpose of studying fighter agility, its modularity should make it easily adaptable to the analysis of other technologies and, thus, be of interest to a number of potential users.

Nomenclature

AML	- Adaptive Maneuvering Logic
AML'	- modified version of COSMIC AML
COSMIC	- Computer Software Management and Information Center
DMS	- Differential Maneuvering Simulator
d.o.f.	- degrees-of-freedom
SA	- Superagility
TDG	- Tactical Decision Generator
TGRES	- Tactical Guidance Research and Evaluation System
TMS	- Tactical Maneuvering Simulator
TOF	- Time on Offense
WVR	- Within Visual Range

Introduction

As new technologies or capabilities are proposed for inclusion in high-performance aircraft, it is imperative to assess the utilization and impact of these technologies within the context of air-to-air combat. This assessment can be a difficult task, particularly, when, faced with technologies which are well outside the range of current experience. A case in point is Superagility (SA). SA is defined as enhanced maneuverability and controllability throughout an expanded flight envelope and may be achieved through the blending of conventional and advanced control effectors (e.g., thrust vectoring, forebody strakes) using a highly integrated control system. SA facilitates unconventional maneuver options like post-stall maneuvering (i.e., supermaneuverability), rolls about the velocity vector under load, and nosepointing. Of course, this new capability is not without drawbacks as well. Designs will be more complex with potential increases in size, weight, and cost and decreases in other measures of performance. In addition, the increased physical and mental stresses placed on the pilot by SA may limit the realizable performance of the man/machine system.

Herbst[1], Foltyn[2], Pennington[3], and others have tried to demonstrate the utility of various aspects of SA through theoretical analysis and simulation. While this work has done a great deal to reveal the potential of SA, controversy still remains about its use and impact in tactically meaningful scenarios. To date, many of these studies have been restricted to simplified airplane models (e.g., point mass). While perfectly valid results can (and should) be obtained from such analysis, in an area such as SA, where the non-linear effects of large amplitude maneuvers often have first-order effects, these results need to be validated through more sophisticated analysis. Currently this analysis is frequently left undone due to lack of a proper simulation tool; existing tools are either oversimplified or not suitable for the task. Also, most studies have been limited to a single airplane or a duel between opposing aircraft because an applicable means of guiding multiple aircraft during within visual range (WVR) combat has not existed for batch simulations, and prohibitively large numbers of runs and pilots are needed to obtain statistically meaningful results in real-time simulations. These limitations have left unanswered concerns that the tactics and benefits of SA seen in one-versus-one engagements may not carry over into engagements with more than two aircraft. Similarly, lack of suitable analysis tools has restricted the study of trade-offs between SA and other potentially conflicting characteristics such as observability--thus, preventing SA from being evaluated in a comprehensive manner.

To help address these issues, NASA Langley Research Center is developing the Tactical Guidance Research and Evaluation System (TGRES, pronounced "tigress"). TGRES is similar in intent to existing digital air combat simulations such as the Adaptive Maneuvering Logic (AML)[4], TAC Brawler[5], or the Air-to-Air Systems Performance Evaluation Model (AASPEM)[6], enabling the study of airplane combat tactics or systems in a realistic but controlled and repeatable environment. These systems allow the researcher to define the initial conditions of an air combat engagement by specifying airplane models, tactical doctrines, guidance logics, and initial airplane states. Using a batch simulation, that includes the potential for occasional operator intervention, the engagement evolves using the specified rules and constraints. By performing a parametric analysis over one of the input quantities, the effect of this quantity can be evaluated. TGRES differs from existing systems by capitalizing on developments in computing techniques to increase system capability and flexibility, and by future incorporation of six-degrees-of-freedom (6 d.o.f.)

[*] Research Engineer, Guidance and Controls Division
Member AIAA

[+] Research Engineer, Guidance and Controls Division airplane dynamics. An additional feature of the system is the inclusion of Langley Research Center's Differential Maneuvering Simulator (DMS), providing the ability to include experienced pilots in real-time evaluations.

This paper will describe the development, current status, and future plans of TGRES.

Overview of TGRES

TGRES consists of three main elements, the Tactical Decision Generator (TDG), the Tactical Maneuver Simulator (TMS) and the DMS as shown in figure 1. The TMS and DMS provide simulation environments, while the TDG is a knowledge-based guidance system exercised within these environments. The intended use of the system is to systematically determine the benefits of SA and to develop tactics which exploit these benefits. This is being accomplished by first developing and analyzing tactics in a batch simulation mode using the TDG and the TMS.

Each time a change or enhancement is made to the ₁DG, a series of 64 engagements with varying initial conditions are simulated using the TMS. The effects of the change are evaluated by studying the resulting trajectories and statistics compiled during each engagement. These statistics include first shot opportunities, time on offense, probability of survival and accumulated time with the opponent in the gun and missile envelopes. After the effects of a change to the TDG are understood, additional modifications are made and evaluated in a quasi-optimization process to maximize the benefits of a given change. After the logic has converged to a stable configuration, a more comprehensive set of initial conditions are run and the results are evaluated. If the results of these batch simulations show the expected level of improvement, the modified guidance logic is then evaluated in manned simulation in the DMS.

A series of engineering and test pilots are used as opponents to fly against the TDG in real-time simulation. Each pilot completes a standardized sequence of initial engagement conditions including advantaged, disadvantaged, and neutral positions. The results of these engagements are evaluated using the same measures of performance used in the batch analysis, with the addition of pilot comments and suggestions. Candidate modifications arising from the manned simulations are incorporated into the TDG and the

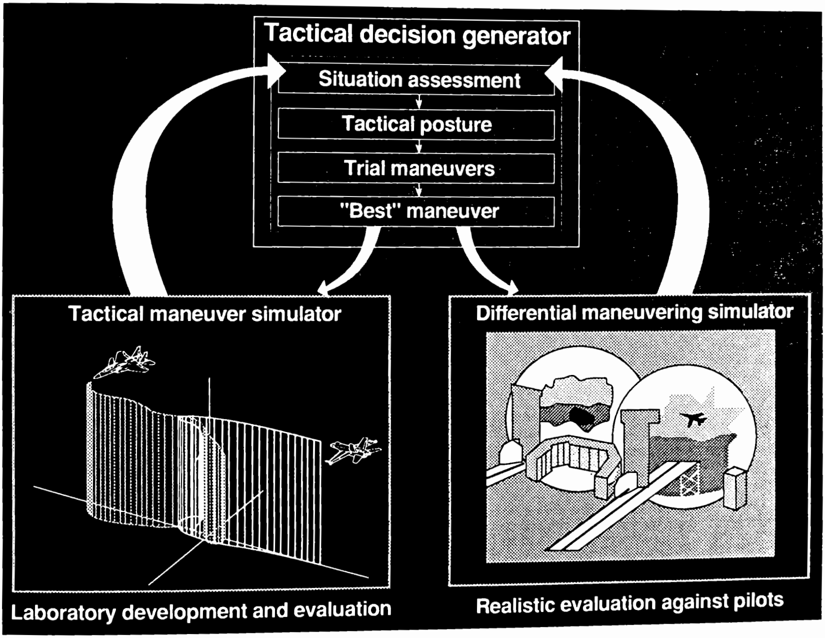

Tactical decision generator
Situation assessment
Tactical posture
Trial maneuvers
"Best" maneuver
Tactical maneuver simulator
Differential maneuvering simulator
Laboratory development and evaluation
Realistic evaluation against pilots

Figure 1 - The three elements of TGRES.

batch and manned evaluation processes are repeated. The net result of this research will be a heightened understanding of the fundamental characteristics of SA and how these characteristics can be exploited to achieve maximum effectiveness.

The modularity of TGRES is such that the elements of the system can be implemented in computationally favorable environments. The TDG is being developed and implemented primarily on a Symbolics 3650® using Lisp as the programming language. The TMS is being developed and implemented primarily on a DEC VAX Station3200® using conventional programming languages. The DMS currently uses a CDC Cyber 175® mainframe computer and FORTRAN as the programming language. The elements execute as co-tasks and communicate via network connections. In addition to enhancing development and implementation efficiency, the task-tailored computational environments provide a degree of parallel processing which will result in reduced execution time.

The following three sections describe the individual elements of TGRES.

Tactical Decision Generator

The TDG is a knowledge-based guidance system programmed in Lisp and designed to explore the maneuvering options offered by SA. It currently utilizes the trial maneuver concept developed for the AML[4] with several extensions based on Artificial Intelligence programming techniques. The trial maneuver concept involves predicting for a given decision interval, the future position, attitude and velocity of the opponent by extrapolating along a second-order curve fit to the opponent's three last known positions. A series of trial maneuvers are then integrated forward in time to determine potential future locations of the guided airplane. Currently, the trial maneuvers employed by the TDG are 'elemental maneuvers', consisting of combinations of load factor, bank angle and thrust. A set of metrics defining a situation space is used to evaluate the worthiness of the elemental maneuvers and the highest scoring maneuver is then executed. The trial maneuver process, shown schematically in figure 2, enables the TDG to construct complex trajectories similar to those produced by human pilots.

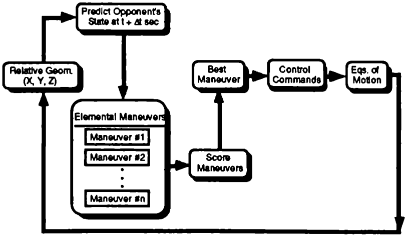


Figure 2 - Trial Maneuver Process

The TDG differs from the AML (as obtained from the Computer Software Management and Information Center, COSMIC) in having a more sophisticated evaluation logic. Whereas the AML has a fixed set of 15 binary "yes/no" tactical evaluation metrics or questions defining the situation space, the TDG uses "fuzzy logic" to more accurately define the space. For instance, to the question, "Is the opponent within my weapons envelope?", AML is restricted to either a yes or no (1 or 0, respectively) response. The TDG assigns a value between zero and one depending on the degree to which the opponent is in the envelope. A near zero value is assigned if the opponent is on the edge of the envelope and is increased until reaching a value of one as the opponent occupies the heart of the weapons envelope and the probability of kill is high.

The TDG also uses distinct modes of operation to vary the relative importance of the individual metrics in response to a change in situation. The TDG contains a situation assessment module which determines the mode of operation by evaluating the state of its own airplane, its current mission or objective, the relative geometry of the opponent, and the opponents instantaneous intent. Each mode of operation has a unique set of weights which are applied to the individual scoring metrics, adjusting their relative importance. The modes of operation currently supported by the TDG are:

- Aggressive
- Defensive
- Neutral
- Bugout
- Evasive
    - missile evasion
    - evade weapon lock of opponent
- Ground/stall avoidance

In addition to a unique set of scoring weights, each mode has a task-tailored decision interval length. For example, while in the aggressive mode, the TDG is frequently fine tracking its opponent. During this tracking it has been found that a relatively short decision interval time (~ 0.5 sec) results in a more robust solution. During more neutral maneuvering, this same short decision interval time results in the TDG performing a "thrashing" motion, unnecessarily bleeding energy, and lowering effectiveness. Thus, during the neutral mode, a longer interval (~ 1.0 sec) is employed.

The TDG is currently being exercised in both batch simulaticn and in real-time simulation against pilots using the airplane dynamics contained in the AML. The airplane model utilized from AML is a performance model, meaning that the thrust, lift, drag and turning capabilities are represented by their maximum, steady-state values. Thus, the transient behavior of the airplane is not accurately modeled. During these exercises, data representing the characteristics of a modern high-performance fighter were used in the airplane models. Figure 3 summaries the results of a series of 64 engagements, each 100 seconds in duration, between the TDG and a modified version of AML known as AML'. AML' is the version of AML which can be obtained from COSMIC, with the corrections and modifications described in reference 7.

The Time on Offense (TOF) is defined as the accumulated time during which a weapons lock is maintained against the opponent. The ΔTOF is computed as:

$$\Delta \mathrm{TOF} = \mathrm{TOF}_{\mathrm{TDG}} - \mathrm{TOF}_{\mathrm{AML'}}$$

Figure 3 shows that over a comprehensive set of initial conditions, the TDG accumulates more weapons time than does the AML'. Results of the real-time simulations are presented in a subsequent section.

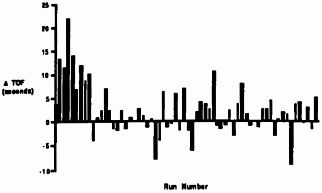


Figure 3 - Performance of TDG versus AML'

As the capability becomes available through the TMS to simulate engagements with three or more airplanes, airplanes capable of performing superagile maneuvers, and weapon flyouts, the focus of the TDG development effort will center on adapting the system to perform in this more complex simulation environment. Enhancements to the TDG's logic must be made to coordinate the efforts of cooperating aircraft as well as considering the threats presented by multiple hostile aircraft. Additionally, the tactical evaluation metrics will need to be further refined to reflect the impact of SA. The current evaluation metrics tend to focus on locally optimum maneuvers; for example, what should be done in the current decision interval to bring the opponent into the weapons envelope. SA will significantly increase the number of maneuver options and while many of these new options may appear to be of immediate benefit, the long-term consequences of these maneuvers (such as extremely high energy loss) may make them ineffective or even suicidal. The TDG needs to recognize and avoid these dead-end maneuvers.

To overcome challenges caused by the increased complexity of the simulation environment, the maneuver selection logic will be expanded to replace the use of trial maneuvers for modes of operation where conventional guidance algorithms provide better performance. For example, during fine tracking, an analytical pursuit algorithm is much more effective at pointing the airplane at an opponent than are trial maneuvers. In addition, the development of mode-specific maneuver sets will increase the effectiveness of the TDG by allowing deeper analysis of situationally appropriate maneuvers during a given decision interval.

The TDG currently uses the method described by Burgin[4] to integrate trial maneuvers forward in time. This method involves making a simplified prediction of the future position of the airplanc by assuming the trajectory to be a segment of a circle corresponding to the normal acceleration characteristic of the particular trial maneuver being analyzed. The effects of thrust and drag are not included in the prediction. This method was originally employed to minimize the computational expense of the trial maneuver concept and yields adequate performance in the context of one-versus-one, conventional air combat maneuvering. However, with the increased emphasis on transient maneuvers introduced by multiple opponents and SA, it is expected that in the environment provided by the TMS, this technique will no longer be sufficient. The reduction in the number of trial maneuvers to be evaluated, made possible through mode-specific maneuver sets, will allow a higher fidelity technique to be used in determining the future position of the airplane. While using a full 6 d.o.f. model for this purpose may still be too time consuming, it should be possible to use a reduced-order, 5 d.o.f. model to accurately predict the future position at an acceptable computation expense.

Tactical Maneuver Simulator

When completed, the TMS will provide a high-fidelity, batch air combat simulation environment in which to develop and test various guidance strategies. The TMS will allow researchers to define the initial conditions between multiple aircraft and then control the trajectories and attitudes of the aircraft using simple trajectory commands or through a guidance system such as the TDG. The TMS will consist of three subelements: a high-fidelity simulation, a tactical autopilot, and a user interface with color graphics and keyboard/mouse control.

The high-fidelity simulation subelement provides for the simulation of the engagement participants and any missiles which may be inflight. Missile flyouts will be simulated rather than using a simpler missile launch envelope analysis as is currently done[7], due to the increased importance of missile evasion maneuvers on the outcome of engagements with multiple opponents. Airplane dynamics are being modeled using the simulation to be described in a forth coming NASA TM. The simulation model is a full, non-linear six d.o.f., rigid-body dynamic model. The aerodynamic forces and moments are calculated from a large wind-tunnel-derived database using table look-ups with linear interpolation. The angle-of-attack range is [-10, +90] degrees and the sideslip range is +/- 20 degrees. In the engine model, the throttle-commanded steady-state thrust level and the dynamic response characteristics of the engine are based on airflow rate as determined from a table look-up. A model of a two-vane thrust-vectoring system is also included and may be optionally activated. This simulation uses the Advanced Continuous Simulation Language[8] as the model-building framework. The user of the TMS will be able to specify the number, type (thrustvectoring or non thrust-vectoring), and initial states of the engagement participants. While the TMS is being structured to allow any number of participants during a batch simulation, more than three aircraft may make the system unacceptably slow while executing on a single VAX workstation. A more powerful machine or additional processors could be used to improve performance.

Unlike most existing batch, air combat simulations, the airplane dynamics in the TMS will be modeled using a full six-degrees-of-freedom. The large amplitude maneuvers typical of air combat maneuvering, combined with the unusual attitudes enabled by superagility, make the use of 6 d.o.f. almost mandatory to accurately model the transient motion of the airplane. However, the use of 6 d.o.f. will place additional, complicating demands on the guidance element (in this case, the TDG) unless an interface is added to deal with these new degrees-of-freedom.

In the TMS, the tactical autopilot will provide this interface. The intent of the autopilot is to allow the guidance logic to operate with the minimum number of degrees-of-freedom required to specify the desired airplane trajectory, thus, freeing the guidance logic from the burden of "flying" the 6 d.o.f. airplane simulation. The function of the tactical autopilot might be compared to the "piloting" actions of a fighter pilot in the sense that it translates the trajectory and attitude commands issued by the guidance logic into control system inputs which cause the airplane to follow the desired path. The placement of the autopilot in the TDG/simulation model system is shown in figure 4. The location of the autopilot is intended to minimize the modifications needed to implement different control systems. It should be possible to substitute candidate control systems of the same family (i.e., rate command, attitude command, etc.) into the simulation without modifying the autopilot, thus, providing a means of comparing in batch simulation the impact of the control system on the tactical effectiveness of the airplane.

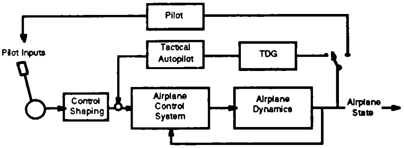


Figure 4 - Location of tactical autopilot

For most conventional air combat maneuvering, coordinated flight (i.e., minimum sideslip) is desired. Trajectories of this nature can be described by the guidance logic using bank angle, load factor, and thrust as outputs. An approach similar to that described in reference 9 is being used to design the tactical autopilot using these quantities as inputs. For unconventional maneuvering, it is frequently desired to command the attitude or attitude rate of the airplane. For instance, it may be possible to achieve a weapons lock by slewing the nose of the airplane to point directly at an opponent. A mode which can be selected by the guidance logic is also being designed into the autopilot to support this type of maneuvering.

The user interface under development permits the TGRES user to examine the results of previously simulated engagements and to interact with engagements as they occur using a color graphics system with mouse and keyboard inputs. The engagement replay system, shown in figure 5, allows the user to replay data recorded during engagements simulated using the TMS and DMS. The replay system plots the trajectories of the engagement participants on a three-dimensional axis system which is surrounded by windows displaying the instantaneous and maximum values of key airplane quantities such as altitude, Mach number, and deviation angle from the opponent[7]. In addition to the replay system, the interactive system enables the user to change the status of the TDG-controlled airplane's systems using a mouse-sensitive display. This capability allows the user to study the response of the guidance logic to changes in system status. Future enhancements to the user interface will include the ability for the operator to manually input guidance commands directly for one or more of the airplanes, bypassing or overriding the guidance logic. By allowing this manual control, the TMS becomes a non-real-time, man-in-the loop tactical workstation. Experienced pilots and tacticians could then use TGRES to interactively evaluate and improve the guidance logic. Engagements would be run under the control of the guidance logic, but with the supervision of an experienced tactician. If the logic performed a maneuver that did not seem tactically correct, the operator could go back to the start of the maneuver and manually fly the airplane through the sequence, trying different options. After determining an effective maneuver or counter, the operator would then modify the guidance logic to reflect the experience gained during the engagement.

Differential Maneuvering Simulator

The third element of TGRES is the DMS. The DMS is a twin dome (40' in diameter) simulator located at Langley and is intended for real-time simulation of engagements between piloted aircraft. By using the TDG to "pilot" one of the simulated airplanes, the decision logic can be evaluated against an unpredictable and highly adaptive human opponent in realtime. This capability has proven invaluable during the development of the TDG.

As was described in the section on the TDG, the development of the decision logic is done primarily in batch using the TMS. While the repeatability of the batch simulation provides an excellent environment for developing the decision logic, this same repeatability can cause difficulties. During the batch development process a baseline decision logic is generally used as the opponent. This opponent performs tactically sound maneuvers and within the constraints of its programming adapts to the maneuvers of the attacker. However, as the decision logic is constrained to proven tactics, it is possible for the logic to overlook an improbable, but effective maneuver. By using a number of human pilots as opponents in the DMS, these 'holes' in the decision logic are uncovered and eliminated by modifying the TDG. When the TMS is completed, much of this work can be done using the tactical workstation feature, but even with this capability, the real-time simulator will remain a vital part of TGRES due to the natural interface the DMS affords to experienced pilots.

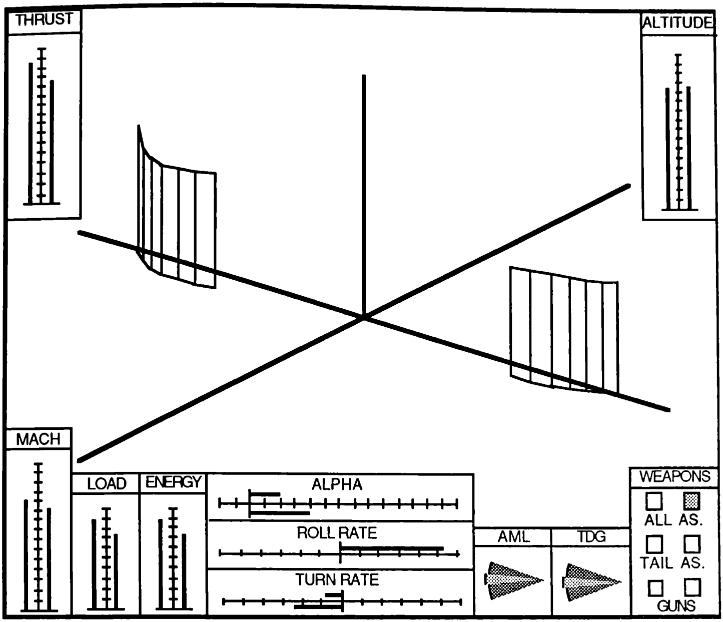


Figure 5 - TGRES Engagement Replay System

Completing three runs at each of the initial conditions shown in figure 6, the TDG slightly outperformed an experienced engineering pilot, achieving an exchange ratio advantage of about 10 percent. The overall exchange ratio is calculated as:

$$\mathrm{OER} = \frac{\text{\# Piloted airplanes disabled}}{\text{\# TDG-controlled airplanes disabled}}$$

Participants are considered disabled when their probability of survival falls below 30 percent. The pilot used in the collection of this data has the highest experience level flying against the TDG. The engagements were started at the edge of the WVR envelope with the pilot being able to see the position of the TDG-controlled airplane but often not being able to determine the attitude of the airplane until observing its motions after the start of the simulation. The initial conditions are biased such that the pilot has an overall position advantage equivalent to an exchange ratio of about 0.75. The overall exchange ratio over 27 engagements of .83 leads to the conclusion that the TDG slightly outperformed its human opponent[7].

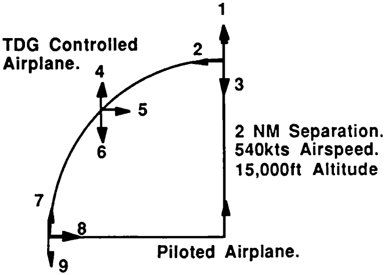


Figure 6 - Initial conditions used in DMS

Future plans for the DMS include the addition of a third dome (20' diameter), enabling the simulation of engagements composed of up to three pilots or airplanes controlled by guidance logics. This addition will allow the TDG to be evaluated in real-time, one-versus-two scenarios, further enhancing the tactical realism and utility of TGRES.

Concluding Remarks

The completed TGRES will provide an integrated environment for developing tactical guidance algorithms and evaluating the impact of new technologies. While being developed to study and learn how to design for SA, the modularity of the system will make it possible to easily modify or replace an element of the system in order to appraise other technologies. TGRES differs from existing systems by capitalizing on Artificial Intelligence programming techniques in the implementation of the guidance logic. The TDG, the guidance element of TGRES, has been shown to perform significantly better than the AML' in WVR engagements. The system will also include a high-fidelity, batch simulation module capable of accurately modeling airplane dynamics, including control system effects, throughout the flight envelope. It will be possible to assess the impact of these dynamics in tactically significant batch and real-time simulations, thus, enabling dynamics optimization throughout the airplane design process. In addition, the tactical workstation feature and the DMS facilitate natural interaction with tacticians and pilots, allowing the decision logic to be tested in a realistic environment. This capability has resulted in significant improvements to the decision logic by exposing deficiencies that would have been difficult to detect using purely batch simulations.

References

1. Herbst, W. B.: Supermaneuverability. MBB/FE1/S/PUB/120, 1983.

2. Foltyn, R. W., et al.: Development of Innovative Air Combat Measures of Merit for Supermaneuverable Fighters. Eidetics Int., TR 87-211, 1987.

3. Pennington, J. E., and Meintel, A. J., Jr: Simulation Study of Thrust Vectoring for Air Combat Maneuvering for a Lightweight Fighter-Class Aircraft. NASA TM X-3270, 1975.

4. Burgin, G. H., Fogel, L. J. and Phelps, J. P.: An Adaptive Maneuvering Logic Computer Program for the Simulation of One-on-One Air-to-Air Combat. Vol I NASA CR-2582, 1975.

5. Kerchner, R. M., et al.: The TAC BRAWLER Air Combat Simulation Analyst Manual (Rev. 3.0). Decision Science Applications Report #668, 1985.

6. Advanced Air-to-Air System Performance Model (ASSPEM) Users Manual, Boeing Document D180-28938-1, November 1985.

7. McManus, J. W., and Goodrich, K. H.: Application of Artificial Intelligence (AI) Programming Techniques To Tactical Guidance for Fighter Aircraft. AIAA Paper 89-3525, August 1989.

8. *Advance Continuous Simulation Language (ACSL) Reference Manual, Fourth Edition.* Mitchell and Gauthier Associates, Concord, MA, 1986.

9. Burgin, B. H., and Eggleston, D. M.: Design of an All-Attitude Flight Control System To Execute Commanded Bank Angles and Angles of Attack. NASA CR-145004, 1976.

ONE ON ONE HELICOPTER COMBAT SIMULATED BY CHESS TYPE LOOKAHEAD

Amnon Katz, Ph.D.* and Arthur Ross, Ph.D.+
McDonnell Douglas Helicopter Company

ABSTRACT

This paper describes an implementation of one versus one air combat logic for helicopters based on classical game theory. The importance of terrain in the rotary wing application suggests the use of a lookahead technique. the details of adapting this technique to a continuous game are discussed. Results of engagements between identical combatants are presented. The effect of compute time on the results is studied.

1. INTRODUCTION.

This paper describes a one versus one air combat logic based on a chess type lookahead technique and presents results of a computer implementation of same.

We follow a well established technique of approximating a continuous differential game as a sequence of selections from a small set of maximum load maneuvers (refs 1,2,3). The observation that air combat normally proceeds through maximum load maneuvers may help explain why such an approximation can be adequate.

The relative position of two aircraft may also be aproximated in terms of discrete cells. An optimal strategy for combat can be developed off line in terms of these. However, in the case of helicopters, which fly and fight in nap-of-the earth, the details of terrain form part of the situation. In this case it is no longer practical to list all possible situations, and, as in chess, the lookahead approach becomes preferable.

Terrain is the reason for employing the lookahead technique, but the present study does not include the effects of hilly terrain. It is intended to explore the details of the lookahead technique itself and its effectiveness.

The lookahead approach has been applied to air combat by Austin et al (ref. 4). The implementation presented here differs from the previous art in:

1. The decisions of the players are staggered rather than simultaneous.

2. All scoring and propagation of scores is done within the framework of the theory of probability.

Some of the principles of the approach above have been suggested in ref 5, but to the best of our knowledge same have never been implemented before.

We present ıesults of engagements between identical advcrsaries both employing the present logic. The engagements are first computed without allowing for the delay required for computation of a decision, then with this effect taken into account. The effect of time of computation on the effectiveness of the logic are discussed.

2. LOOKAHEAD IN AIR COMBAT

Following previous art, we allow the combatants discrete decisions of selecting among a small number of maximum load maneuvers. In the present study we allowed nine maneuvers, which are listed in Appendix A. Once selected, a maneuver is maintained until criteria for a new decision (to be discussed presently) are met.

We use the resulting game tree to propagate scores and let Blue and Red select the branches leading to the highest and lowest scores respectively.

Unlike chess, kill in the air combat tree is probabilistic rather than deterministic. Also it occurs along a branch rather than at a node. For this reason the discrete score function of chess (with values 1,0,-1 for win,draw,or loss) is replaced by the expectation value of the corresponding variable in air combat:

$$V = \begin{cases} 1, \text{ if Red is killed and Blue survives,} \\ -1, \text{ if Red survives and Blue is killed,} \\ 0, \text{ otherwise.} \end{cases} \quad (1)$$

$$S = \langle V \rangle. \quad (2)$$

Blue's goal is to maximize S, Red's to minimize it. S is limited to the range [-1...1].

Propagation through the game tree requires estimates for the following probabilities for each branch:

P_b - the probability that Blue is killed,

P_r - the probability that Red is killed, and

P_s - the probability that both survive.

Armed with these it is possible to propagate the score S along the branch from the node at the i+1 level to the node at the i level by

$$S_i = P_r - P_b + P_s S_{i+1}. \quad (3)$$

If the node i corresponds to Blue's decision, the highest propagated score and corresponding branch are selected. The lowest score is selected at Red's decision.

We estimate the probabilities based on time spent in the opponent's gun envelope. The details are

*Manager, Advanced Development Office, Simulation System Group. Member AIAA.
+Advanced Dev. Office, Simulation System Group.

presented in Appendix B. When terrain is taken into account, time spent too high and out of cover may be similarly penalized.

It must be stressed that this probabilistic formalism does not represent the kill mechnism of the simulated engagement. In the present study there is no kill mechanism. We merely observe how the combatants persist in their efforts to bring the opponent into the gun envelope. Conversely, one could in principle simulate individual or grouped projectiles and determine kill based on hit or miss. The probabilities above serve only for the thought processes of the adversaries - i.e. for the lookahead process.

The score at a terminal node is a heuristic estimate of <V> based on the conditions there. The form used for the present study

$$S_f = \tfrac{1}{2}[\cos(Ab) - \cos(Ar)], \qquad (4)$$

where Ab is the angle between Blue's body axis and his line of sight to Red, and Ar is the angle between Red's body axis and his line of sight to Blue.

This form takes account of relative orientation only, and rewards each player for facing his opponent. The results show that it leads to tenatious players who always turn towards their opponent.

Another aspect of the air combat tree that differs from chess is that a specific time is associated with each node. The time is not necessarily the same for nodes at the same depth.

The driving consideration here is the need for the lookahead to proceed considerably faster than real time. (Failing that, the term lookahead would be a misnomer.) Yet the combinatorial explosion of the tree exponentially slows its progress as it proceeds from one level to the next. It is therefore necessary to limit the tree search to very few steps, and have these steps cover considerably longer than the time required to compute them. This precludes the use of very short steps.

Yet if held too long, a maneuver may outlive its potential benefit and any advantage it may have provided could be lost. A turning maneuver may improve a player's position by aligning his gun envelope with the opponent. But if held after this is achieved, the advantage would be lost.

The solution is to determine the duration of each maneuver by estimating the time by which it achieves its effect. This could take the form of following the progress of the score function S and create a new node when it peaks - that is when it reaches a local extremum.

The result is that the branches of the air combat tree (Figure 1) are unequal in length. Figure 1 shows a tree in which decisions are taken by the players in turns.

With alternating decisions, each node must be created in advance of the time when the current maneuver is played out.

It is equally possible to let both sides make their decisions simultaneously. That is the approach taken by Austin et al. On the face of it both approaches tend to the continuous game as the decisions become dense. However the nature of such a limiting process could be profoundly affected by this detail (compare the limiting process in representing quantum mechanics by functional integrals). The effect of staggerering decisions on the limiting process in the present case is not known.

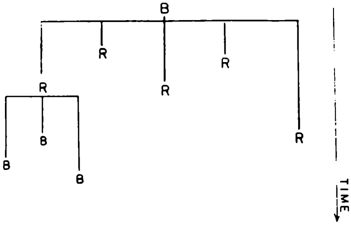


Figure 1: Air Combat Tree with Alternating Decisions

Please note again that the discussion of staggering of decisions here does not apply to the decisions actually made. (See below; they were simultaneous.) The discussion applies only to the decisions in the lookahead tree, that is the decisions that each combatant "imagines" himself and his opponent reach in the future, as he seraches for his current decision.

3. RESULTS FOR INFINITELY FAST LOGIC

In this section we discuss engagements computed between identical combatants both employing the combat logic discussed here. The simulated engagement made no allowance for the time it took to compute a lookahead decision.

The combatants made their decisions simultaneously. Starting with their initial condition, each constructed a game tree (with staggered decisions) and based on it reached a decision. No allowance was made for the time this process lasted - the decisions were made available to the adversaries while still at their initial position. Each then implemented his decision until it was time for the next decision.

The following parameters applied:

The lookahead tree contained two plies (one level for the decision maker and one for his adversary) and reached up to 4 seconds into the future.

A new tree was constructed, and a new decision reached at intervals of one second.

Low level logic supplied control inputs to the flight model at intervals of 0.1 second. These inputs were calculated to maintain the maneuver last selected.

In the absence of varying terrain, the laws of motion and of decision of the players are invariant under a group of transformations consisting of:

1. Horizontal translations.
2. Rotations around a vertical line.
3. Reflection in a vertical plane.

Given an intial condition that is invariant under a transformation belonging to the group above, all subsequent situations as the engagement unfolds must maintain the symmetry. With the players being identical, the situation is invariant under transformations that merely interchange the players. This provides a powerful check. Deviations from symmetry betray errors in logic or coding.

Three types of symmetry were sampled:

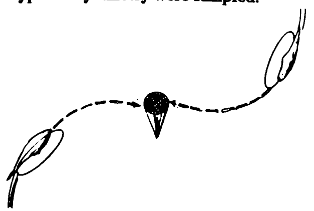


Figure 2: An engagement symmetric in reflections through a vertical line (shown as a pole). Perspective view.

I. Symmetry in reflection through a vertical line. (Same as rotation of 180 degrees around a vertical line.) (Figure 2):

These engagements remained in the vicinity of the a vertical line through the point of symmetry with the combatants descending or climbing together while constantly turning into each other and aiming for a collision course. No collision avoidance was included in the combat logic. No effects of collisions were incorporated in the flight dynamics. Collision courses were prefered because they put the adversary in the center of the gun envelope.

The players repeatedly grazed each other or even flew through each other. The amount by which they missed a precise collision course was the result of the granularity of the available maneuvers and decision times.

The players descended together, maintaining equal altitudes and symmetry in reflections through the vertical line through their initial point of symmetry.

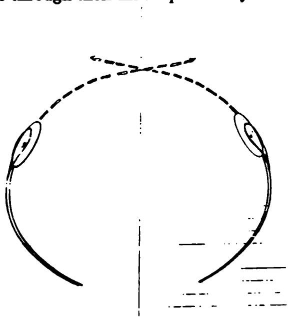


Figure 3: An engagement symmetric in reflections in a vertical plane. Top view.

Each player strove to keep the other in the gun envelope. Needless to say that neither side gained an advantage. Both appeared tenatious and aggressive - as one would expect and desire.

II. Symmetry in reflections through a vertical plane (Figure 3):

These engagements remained in the vicinity of the plane of symmetry and proceeded along the plane. The players kept turning into each other and flying through each other at the plane of symmetry. The battle proceeded horizontally one way along the plane then tracked back the other way. Again neither player could gain an advantage. Both appeared aggressive and tenatious.

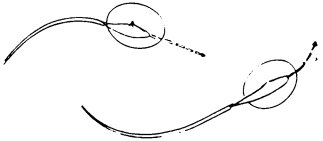


Figure 4: An unsymmetric engagement. Perspective view. The players are identical, but the initial and subsequent conditions are not symmetric.

III. Unsymmetric initial conditions: (Figure 4)

In this case the combatants were started at different speeds and altitudes. They employed different tactics. At one point one player gained a position on the other's tail and, for a while, maintained it by what looked like scissoring.

4. THE EFFECT OF COMPUTE TIME.

In any application of combat logic to either combat or real time simulation, account must be taken of the time it takes to compute a decision. The computation cannot start until the initial position is known. By the time the decision is made, the compute time has elapsed, and the condition assumed in the computation no longer prevail.

We make the simplifying assumption that the time to compute a decision is equal to the time interval between decisions. This should obtain when the system performing the lookahead computation is dedicated to this task and is ready to start a new lookahead as soon as one is concluded.

We also assume that this compute delay is constant and known in advance. The given initial conditions may be projected ahead by the same amount of time, and the decision computed for the conditions expected to prevail at the time it becomes available.

As applied to the the parameters of section 3, the compute delay is taken to be one second. This effect was simulated by applying the lookahead to the last step conditions as projected for the duration of one step rather than the current step conditions.

The three symmetry cases of Section 3 were redone with the delay accounted for. The symmetry check still applies. The delay is reflected in a slower reaction on the part of the players. It takes them about a second longer to start turning towards each other once they had passed each other. The engagement is not as tight as it was, and the area over which it unfolds is increased.

Certain pathological cases also arise, where one or both players move away and fail to turn back. instead they alternate right and left turns. The maneuver appropriate for the opponent's decision before last is always reversal of the current turn. This can happen with the opponent directly behind (either following or also moving away) and is dependent on the exact synchronization of the players' decisions. It is unlikely to occur when confronting a human or otherwise different opponent.

Apart from this pathology, the players are unchanged in nature. They are still aggressive and tenacious.

5. SUMMARY

In summary we find that the lookahead is a viable approach for air combat and yields impressive results with very simple minded tools. This is not unusual in applications of a practical nature (6). The timing assumed is such as can be easily accomodated by embedded microprocessors. The technique is therefore ripe for use in real time simulation.

APPENDIX A: LIST OF MANEUVERS

This study allowed only 11 maneuvers:

1. Straight and level flight at constant speed.

2. Accelerate: Apply all excess rotor thrust forward. Reach and maintain maximum attainable speed or maximum speed in the allowed envelope, whichever is less.

3. Decelerate. Apply all excess rotor thrust aft. Reach and maintain specified minimum speed.

4 through 11.

These are maneuvers where all excess rotor thrust is applied laterally. In this case the maneuver is coupled with acceleration/deceleration to the speed V0 permitting maximum excess load. If current speed is higher than V0, forward thrust is reduced to whatever is necessary to balance drag at V0. If current speed is lower than v0, the aircraft is first accelerated to V0. The directions of lateral thrust application are shown in Figure 5.

NOTE: Direct control of the flight model is maintained by a low level "maneuver interpretr" which computes control inputs necessary to start/maintain the selected maneuver. These inputs are recomputed at short intervals (0.1 seconds in the current implementation) regardless of whether the maneuver selection was changed or not.

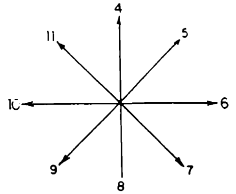


Figure 5: Direction of ecxess rotor thrust in maneuvers #4 through 11

APPENDIX B: ESTIMATE OF PROBABILITIES

This Appendix presents the simple method of estimating probabilities that was employed in the present study.

We assume that the probability of kill over an infinitesimal time dt spent in a lethal environment proportional to dt:

$$dP_1 = C\ dt. \tag{B.1}$$

The probability Po of surviving a period t+dt may be decomposed as

$$P_0(t+dt) = P_0(t)\ (1 - C\ dt), \tag{B.2}$$

which leads to

$$\frac{dP_0(t)}{dt} = \begin{cases} -C, \text{ when in lethal environment,} \\ 0, \text{ otherwise.} \end{cases} \tag{B.3}$$

This integrates to

$$P_0(t) = \exp\{-Ct_L\}, \tag{B.4}$$

where

t_L

is the cumulative time spent in a lethal environment. The probability of kill is:

$$P_1(t) = 1 - P_0(t). \tag{B.5}$$

REFERENCES

1. Falco, M., Carpenter, G., and Kaercher, A. "The Analysis of Tactics and System Capability in Aerial Dogfight Game Models" Grumman Report RE-474 (May 1974).

2. Falco, M., Hinz, H., and Sobierajski, F. "Combat Performance for Light Helicopter Concepts", Grumman Report RE-616 (December 1980).

3. Falco M., "Light Helicopters Air to Air Combat - A Digital imulation Study", Grumman Report RE-683 (April 1984) Confidential.

4. Austin, F., Carbone, G., Falco, M., and Hinz, H. "Automated Maneuvering Decisions for Air to Air Combat", Grumman Report RE-742 (November 1987).

5. Bent, N. E., Densmore, J. E., Dickerson, M. M., and Kerchner, R. M. "Design Plan for Rotor Brawler, an Intelligent Player Module for the MDHC ETSD Manned Helicopter Simulator", DSA Report No. 873, January 29, 1988.

6. Kaindl H., "Minimaxing - theory and practice", AI Magazine Vol 9, Issue 3 (Fall 1988).

7. Katz, A. "Classical Mechanics, Quantum Mechanics, Field Theory", Academic Press, New York (1965)

ELECTRONIC ORDER OF BATTLE IN A REAL-TIME MULTIPLE ENGAGEMENT SIMULATION

Kevin C. Klos*

Northrop Corporation
Aircraft Division
Hawthorne, California

Abstract

At Northrop Corporation's Aircraft Division, the Integrated Systems Simulation Laboratory (ISSL) has developed an advanced interactive simulation of an Integrated Air Defense System (IADS) for use in their multiple engagement tactical simulation. This Electronic Order of Battle (EOB) model provides a representation of the elements, functions, and interfaces that exist in a generic air defense system. EOB interacts with various elements of the real-time simulation to provide a realistic ground threat environment. Data reporting capabilities provide users with information on many aspects of aircraft system performance against the model. The generic nature of the model source code allows development and testing in an open environment. The data driven nature of the model permits projects to tailor the ground threat environment to their specific needs simply through data file changes. Modeling of a specific air defense system is achieved by assigning appropriate operating parameters and characteristics to the various elements of the model.

Introduction

Northrop Corporation's Aircraft Division has an operational real-time simulation facility to provide projects at Northrop with an advanced manned interactive multiple engagement tactical air combat mission simulation capability.[1] One of the significant and unique features of this facility is the real-time Electronic Order of Battle model, a ground based air defense system simulation which interacts with the many other existing simulation models as well as the manned participants.

EOB consists of the following models:

- o a network of Early Warning (EW) radar sites
- o Airborne Warning and Control System (AWACS) aircraft
- o a network of Surface-to-Air Missile (SAM) sites, with associated Fire Control Radars (FCR)
- o a team of manned and computer-controlled Air Interceptors (AI)
- o a man-in-the-loop (MITL) Ground-Controlled Intercept (GCI) station
- o a digital Command, Control, Communications and Intelligence (C^3I) center
- o SAM flyouts
- o initialization routines

The structure of EOB model software is summarized in Figure 1.

The core of EOB was derived from a non-real-time analytical simulation previously developed at Northrop, which contained only non-networked radar sites and a very limited C^3I structure. This simulation was modified for use in the ISSL real-time environment and expanded to include higher-order C^3I functions and weapons modeling.

EOB reacts to aircraft either as friendly, hostile, or neutral. It interacts with the Manned Interactive Control Stations (MICS) and the domed simulators, computer controlled participants, a manned GCI station, the simulation data collection routines, the parametric simulation data display, and the God's Eye View display. Figure 2 shows a pictorial representation of the elements of EOB and a typical multiple engagement simulation scenario.

EW Site Network

The EW site represents the various elements that make up an Early Warning network. In the context of enemy threat air defense, these elements would include Radio Technical Companies

* Engineer II

EW SITE MODEL

Multiple Sites in Beddown
Multiple Radars per Site
Independent Radar Operation
Communication w/ C^{3}I Center

RADAR MODEL

Multiple Radar Types
Operating Parameters
- Sweep Rate
- Antenna Height
- Frequency and Polarity
- High Altitude Capabilities
- Low Altitude Capabilities

Detection Algorithms
- Deterministic
- Probabilistic

Target Radar Cross Sections
- Multiple Patterns
- Function of Radar Frequency and Polarity

C^{3}I CENTER MODEL

Center Tracker
Weapon Allocation
- AI Evaluation
- SAM Site Evaluation

Weapon Assignment
AI Vector Management
SAM Site Management

GCI CONTROLLER

AI Vector Management
SAM Site Management
EW Site Control
AWACS Control

SAM SITE MODEL

Fire Control Radars
Multiple SAM Types
Interface w/ SAMs
Interface w/ C^{3}I Center

SAM FLYOUT MODELS

High Fidelity Missiles
Generic Missiles
- Lead Collision Guidance
- Pursuit Guidance

Endgame Models
- Deterministic
- Probabilistic

AWACS MODEL

Racetrack Pattern on Station
Threat Avoidance Logic
High Fidelity Participant

INITIALIZATION SOFTWARE

Menu Driven Routines
Aircraft RCS Data Files
Radar Parameter Data Files
SAM Parameter Data Files

Figure 1 - EOB Model Structure

organic to both Front and Army commands as well as GCI sites associated with enemy fighter Regiments. Information and decision latency times associated with command and control are modeled and are data driven to allow flexible simulation of different IADS.

The EW site network is EOB's primary source of target information. Up to several hundred EW radar sites may be included in the network, all of which feed target detection data to the C^{3}I Center tracker. Individual EW sites do not communicate, but radars at a single site share target detections.

EW Site Model

An EW site consists of one or more radar types at a given inertial location in the simulation gaming area. Various enemy threat EW/GCI radars have been modeled; radar types are described in a radar model characteristics data file. Sweep rates for each radar type are defined by the radar model, but the EW site model does not simulate a true rotating radar antenna. Individual EW radars perform detection checks against all aircraft every time their 360-degree sweep rate has elapsed. Each radar has its own sweep timer; these timers are set by a random draw during simulation initial conditions time in order to stagger radar activation.

EW Site Tracks. Target tracks at individual radars are established when a given number of consecutive target detections occur, and are lost when a given number of consecutive misses occur. Track establishment and loss criteria are part of the radar operating characteristics and are unique for each radar type.

An EW site itself is considered to have a track on a target when at least one of its constituent radars has a track. EW radars and sites do not actually maintain a track file on each target: radar and site "tracks" consist only of target status and position records, which are fed to the Center tracker for processing. Each radar at each site operates independently in detecting targets, but radars at a single site will communicate and share target detection data. For each EW site, the target position provided to the C^{3}I Center tracker will be that reported by the last radar at the site to register a detection against the target prior to Center track update.

Radar Model

All EW and FCR sites use the same radar detection algorithm. Specific radar types are simulated by initializing them with the appropriate operating characteristics. Many different types of radars can concurrently exist in a single simulation run.

The radar model examines two criteria to determine target detectability: the target's apparent radar cross section (RCS) and the operating characteristics of the radar itself. Both deterministic and probabilistic detection algorithms are available for use in the radar model, allowing projects to tailor the radar model to closely match that of other simulations. Both algorithms use the same radar characteristics data tables for target detection processing.

Target RCS. Aircraft RCS data is tabulated for various radar operating frequencies and polarities; within each frequency and polarity group, RCS values vary as a function of the azimuth and elevation of the radar relative to the target aircraft. Several distinct aircraft RCS patterns can be accessed in a simulation run, allowing the user to fly several different types of threat aircraft in a single engagement. Gain factors are also available for each pattern, allowing the user to increase or decrease an aircraft type's radar visibility proportionally across the entire pattern without needing to store separate data tables.

Radar Characteristics. Radar parameters include operating frequency and polarity, antenna height, sweep rate, and detection capabilities for both high and low altitude targets. High altitude characteristics are defined by pre-processed tables of ranges of 50% probability of detection (Pd) of a 0 dBSM target for various target inertial elevation angles. Low altitude capabilities are given in terms of inner and outer detection ranges based upon target altitude and RCS. Pulse doppler radars can be modeled with a velocity "doppler notch." Targets with a velocity relative to the radar below a given minimum cannot be detected by such radars.

Deterministic Detections. The deterministic detection algorithm for high altitude targets use R^4 scaling to find the 50% Pd range against the target's presented RCS; if the true target range from the radar is less than this range, the target is detectable. For low altitude targets, inner and outer detection ranges are computed as a function of target RCS and altitude; if the true target range from the radar is within these bounds, the target is detectable.

Probabilistic Detections. Probabilistic detections for both high and low altitude regions use the detection tables, plus a probability of false alarm, to compute a probability of detection based on target signal-to-noise ratio which is compared to a Monte Carlo draw for detection determination.

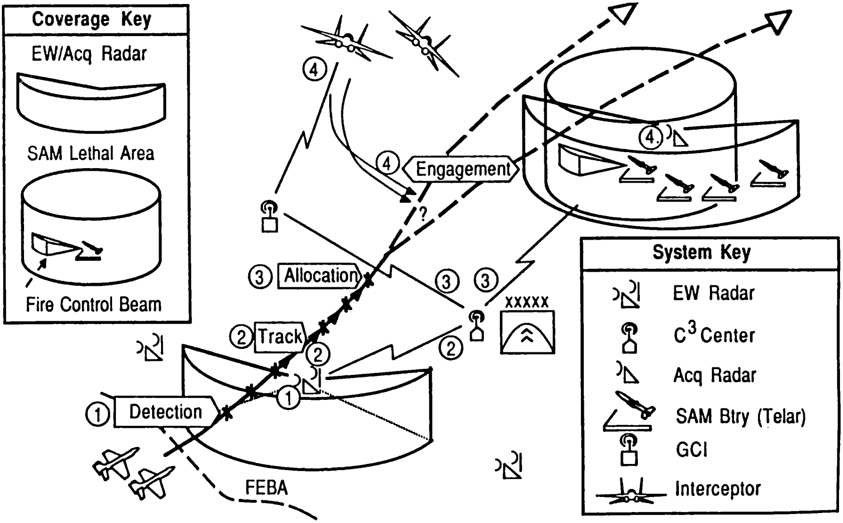


Figure 2 - EOB Engagement Scenario

AWACS Model

EOB AWACS aircraft orbit predefined stctions in an oval "racetrack" pattern. AWACS radar behavior is exactly like that of an EW site; radar characteristics peculiar to a specific AWACS aircraft type are contained in the radar characteristics initialization file. Station-keeping logic is defined by user input parameters. AWACS aircraft will continue to orbit on station until commanded to return to base by the GCI controller or by the C^{3}I Center.

The user has the option of making AWACS aircraft high fidelity participants in the simulation, capable of being detected, tracked, targeted, and killed by the other participants. In this case each AWACS operates station-keeping logic which monitors threats against the AWACS aircraft and commands evasive maneuvers as needed. The user can assign to these high-fidelity AWACS aircraft various levels of desire for self-preservation and desire to remain on station.

C^{3}I Center Model

EOB's C^{3}I Center handles target tracking and weapon allocation and control. The Center represents an enemy Front level air defense command post. Its integrated tracker generates target position and velocity vectors using target detection data from the EW network. It is responsible for the allocation of SAM and AI assets, the assignment of allocated weapons, and the maintenance of AI GCI vectors.

C^{3}I decisions are based upon target detection and tracking information provided to it by the EW site network through the Center tracker, and by a fixed set of rules contained within the model. Center operation occurs at varying frequencies dependent on the status of foe aircraft, friendly air interceptors, and SAM sites, and on user-specified time delays (representing decision time or data processing time) between C^{3}I events. Major time steps occur between initiation of Center track on a target, weapon allocation, weapon assignment, and GCI vector updates.

C^{3}I Center Tracker

The Center tracker actually maintains and updates target tracks based on target detection information fed to it from the EW sites. All sites are networked into the tracker and begin providing target detection data to the Center when EW site track is established, after a user-defined delay period for each site (intended to simulate communications delays inherent in an air defense system) has elapsed. Targets must have a track established at the Center before weapon allocation is attempted.

Track Updates. Center track updates occur at a user-defined rate. At update time, the tracker collects target position data from all EW sites that detected the target since the last Center track update. Detected target position coordinates from each detecting site are averaged and used as input to an alpha/beta linear tracking algorithm, providing perceived target position and velocity vectors.

A user-defined number of Center track updates with valid EW site data must occur on a target before the target is considered to be in track at the Center. Due to the nature of the tracking algorithm used, at least three such updates must occur to allow the tracker to arrive at an initial track solution. Center tracks are dropped after a user-specified number of consecutive updates occur in which no data was provided for the target by the EW site network.

Track Accuracy. The alpha/beta tracker uses an alpha factor, selected by the user, to determine the amount of emphasis put on new EW radar data in updating a track. The base alpha factor is modified in real-time to place more emphasis on new radar data as more EW sites contribute to a track update, and more emphasis on old track data as fewer sites contribute to an update. During updates in which no new EW site data has been provided, existing track velocities remain constant and are used to extrapolate the track position vector.

Depending on the frequency of Center track updates and EW radar sweep rates, site detection data may be somewhat aged at the time it is used by the Center tracker. This causes inaccuracies in track updates which may be increased or decreased by altering the Center track update rate, radar sweep rates, or the tracker alpha factor.

Weapon Allocation

If a target remains in track at the Center for a given amount of time, intended to simulate the delays inherent in managing an air defense weapon network, the Center will attempt to allocate a weapon for intercept. The weapon can be a SAM site, an AI, or both. The user may enable or disable both types of weapons as a group; within each weapon type individual elements may be enabled or disabled as needed, at initial conditions time and interactively in the real-time simulation run.

For each weapon type the element capable of the earliest target intercept, according to the evaluation criteria for that type, will be allocated to the target. Allocated weapons are posted to a "busy" status, so that the weapon will not be considered for allocation against multiple threats, but allocation does not represent a total commitment of the asset to the target.

If no weapon from an enabled type is allocated to a target, the Center will reattempt allocation from that type immediately after the next Zone update of that target's track. This process continues until allocation is successful for each enabled weapon type or until the target drops from Zone track.

Air Interceptor Selection. AI evaluation against target aircraft is an iterative process of calculating a lead-computed intercept point. Current AI heading and airspeed are used to calculate the time it will take the AI to come to the correct heading and airspeed for intercept. Since the time it takes the AI to turn to the correct heading will affect the target's position at intercept, the solution is iterated until errors are minimized.

All non-busy AIs are evaluated in the same way against the target's flight path. An AI is posted as busy if it has already been allocated for intercept, is directed by the GCI Controller to return to his Combat Air Patrol (CAP) station or air base, is low on fuel, or has exhausted his weapons stores. The AI with the smallest projected time-to-intercept is the one allocated for intercept. No indication of Center allocation is given to the AI pilot at this time.

SAM Site Selection. Prior to SAM site evaluation, the target's current track position and velocity vectors are extrapolated to obtain a predicted target position at weapon assignment time. Target course extrapolation is linear, and does not account for target maneuvers. This predicted position is used in evaluating the potential of each SAM site for target intercept.

Several different types of SAMs may be active in a simulation run, each having an acceptable launch region defined by the missile operating characteristics, and an altitude coverage band denoting the SAM type's battlefield zone of responsibility. The target's predicted altitude at SAM assignment time is first checked against the altitude coverage bands for each SAM type; only sites shooting SAM types covering the target's predicted altitude are considered for intercept. Sites already allocated to targets also are not considered for intercept. Multiple target engagement by single SAM sites is not yet modeled.

Of the sites passing the above checks, the one capable of getting a missile to the target in the least amount of time based upon the relative geometry between site and target aircraft is the one allocated to the target by the Center.

Weapon Assignment

If a target remains in Zone track for a user-specified time period after weapon allocation occurs, the Center attempts to assign the allocated weapons to the target. This assignment represents the weapon commitment point.

Before weapon assignment, a final check on intercept capability is made using the same criteria as for weapon allocation. Only the allocated weapon is checked here; non-allocated weapons are not compared for intercept suitability. If the weapon is no longer capable of intercepting the target at assignment time (due to changes in the target's perceived flight path or to weapon limitations) the Center frees the allocated weapon and resumes the allocation process against the target. If the weapon is still suitable for intercept, it is assigned to the target. This is the weapon commit point.

AI GCI Vector Management

When an AI is assigned to intercept a target, the Center directs the AI to the intercept point using symbols projected onto the AI's Head Up Display (HUD). The projected intercept point calculated at assignment time is saved, and a steering dot to this point is projected onto the pilot's head-up display. A marker indicating the target's last know position is also projected onto the pilot's tactical situation display.

Vectors are updated by the Center at a user-specified rate, representing C^3 delays. When vector updates occur, the intercept steering and target position markers are updated to reflect the new intercept solution. If the Center determines that intercept is no longer feasible, the AI is posted to a non-busy status and the GCI cues are removed from his cockpit displays, and the target is scheduled for weapon reallocation as necessary. If the target drops from C^3I Center track during GCI, the assigned AI is directed on a dead reckoning course to the last projected target intercept point. Dead reckoning vectors are canceled by the C^3I Center after the AI reaches the intercept point. Vector updates and cancellations may also be commanded by the GCI controller.

GCI Controller's Station

GCI Station Composition

A color raster-graphics display monitor with a touch-sensitive screen and associated software provides a man-in-the-loop capability to monitor and control asset disposition within EOB.

The GCI station represents the information and authority available at the Front air defense center, and automated command and control tools allow the GCI Controller to act as an air defense weapons operation control commander, not limiting him strictly to the role of a GCI operator.

Target track histories, AI positions and velocities, EW site locations, and SAM site locations are displayed on the screen, along with function buttons which allow the GCI controller to affect the operation of the C^3I model. Various zoom, center, and declutter functions are also available. CAP station locations and the Forward Edge of Battle Area (FEBA) line are also displayed for reference.

AI Control

The GCI Controller's station allows a manned participant to solve tasks which are intuitively simple for humans, but inherently difficult for real-time computer simulations. GCI vector management is one such task. The GCI controller can set the function of his control panel to automatically approve all Center-generated vectors, to require his approval before assigning AI assets, or to generate GCI vectors only at his command. In all of these modes, the GCI controller can at any time generate, update, or cancel any GCI vector. He can also vector AIs to different CAP stations or air bases, and request status on AI stores and fuel.

The manned GCI station also allows advanced technology concepts to be evaluated against established enemy threat MITL C^3I doctrine. Rules of engagement for the GCI Controller can be specified by the user prior to engagement, resulting in a more realistic simulation of a tactical scenario.

SAM & EW Site Control

Control over SAM assignment mirrors that of the GCI vectors: the controller can approve or veto any Center-generated assignments, as well as assign and deallocate specific SAM sites to targets. The GCI controller can also activate and deactivate any EW or SAM site at any time during a simulation run, can move CAP points around the simulation gaming area as needed, and can control the operation of the AWACS aircraft.

SAM/FCR Site Model

At SAM assignment time the Center cues the site's FCR to attempt to detect, track, and fire upon the target. FCR operation is intended to simulate a continuous beam radar. Target detection and tracking is done using the same algorithms as the EW radars: track is established after a given number of consecutive detections, and track drop occurs after a given number of consecutive failed detections. FCRs are modeled as other radars and are SAM type specific.

The FCR model and the SAM flyout models are completely decoupled in their operations within EOB. Handshaking between models allows missile launch and abort requests by the FCR to be processed appropriately by the SAM flyout model, and missile outcomes to be reported back to the FCR model.

SAM Site Operation

SAM sites in EOB currently employ shoot-look-shoot logic against assigned target aircraft. When a FCR develops a track on a target, it immediately requests a SAM launch for the site's designated SAM type. A user-specified time is allowed to elapse after an unsuccessful missile flyout before another launch is requested. Before SAM launch request a check is done against the target to ensure that it is within the engagement envelope of the SAM type installed at the site.

Once a SAM site is assigned by the Center it continues to operate autonomously regardless of the target's status at the C^3I Center. The site will only stop shooting if it has a launcher failure, runs out of available missile stores, loses its FCR track on the target, is de-assigned by the GCI controller, or if the target moves out of the SAM launch envelope. If one of these conditions occur, the site signals to the Center that it has dropped the target. The Center then frees the site for allocation to other targets, and schedules the target for future weapon allocation as needed.

SAM Flyout Models

Both high-fidelity and generic SAM models are available for use in EOB. All SAMs currently modeled are semi-active homing, guided by ground-based fire control radars.

SAM flyouts are updated at the basic simulation frame rate of 20 Hz. Missile position is displayed on the God's Eye View, and also on the MICS/Dome threat warning receivers of those aircraft whose onboard sensors detect them.

High-Fidelity SAM Models

High-fidelity models for several enemy threat SAMs were derived from the Tac Zinger SAMS model and modified for use in the ISSL real-time environment. These missiles consist of a detailed six-degree-of-freedom aerodynamic model, a flight control system with autopilot, and an endgame model. The performance of these missile models has been compared against existing threat data to verify model accuracy. Since these models are data-driven, the performance characteristics can be easily modified to keep pace with updated threat information.

Generic SAM Model

Generic SAMs consist of a simplified five-degree-of-freedom kinematic model, a guidance package, and an endgame model, all of which are data driven. Specific performance requirements are met by assigning appropriate data values to operational parameters such as missile size and mass, thrust and burn time, maneuvering capabilities, and warhead performance. Available guidance models are proportional navigation, lead-collision steering, and line-of-sight (pursuit) steering.

Endgame Models

SAM endgame models can be either deterministic or probabilistic. The availability of both types provides the flexibility to match the operation and performance of EOB SAMs to established off-line SAM flyout models. Perfect missile fusing is assumed; weapon reliability is modeled through degraded probability of kill (Pk) tables. All endgames result in either a miss or a kill; fractional battle damage is not currently simulated.

Both endgame models use missile warhead effectivity data initialized according to SAM type versus aircraft type. These data tables can be a function only of missile miss distance, or of both miss distance and angle of arrival of the missile relative to the target aircraft.

Countermeasures effects in the form of offboard tactical decoys are also modeled: aircraft equipped with the appropriate equipment can eject decoys in response to incoming SAMs. Decoy performance is simulated by a missile Pk degrade factor, which is a function of the elapsed time between decoy dispensing and missile endgame. Optimum dispense times and Pk degrade tables are specific to each decoy/SAM pair.

Deterministic Endgame. Deterministic endgame models make a "cookie cutter" comparison of the final missile miss distance to a 50% Pk range. If the miss distance is inside this range a kill is scored, otherwise the missile misses. Missile lethality can be adjusted by changing this 50% Pk range.

Probabilistic Endgame. Probabilistic endgame models use the SAM miss distance at closest approach as input to a linear interpolator to produce a Pk, which is then compared to a Monte Carlo draw to determine the outcome.

Data Collection

State variables describing EOB model performance and interaction with other simulation elements are taken by both set-frequency and event-driven real-time data collection routines. Figure 3 contains a summary of EOB information recorded during a real-time multiple engagement simulation.

EW SITE NETWORK

EW Site Aircraft Detection Histories
EW Site Track Establishment & Loss

C^3 I CENTER

C^3 I Center Track Establishment & Loss
C^3 I Center Aircraft Track Histories
AI Allocation and Assignment
Sam Site Allocation and Assignment
GCI Controller Activity

SAM SITE NETWORK

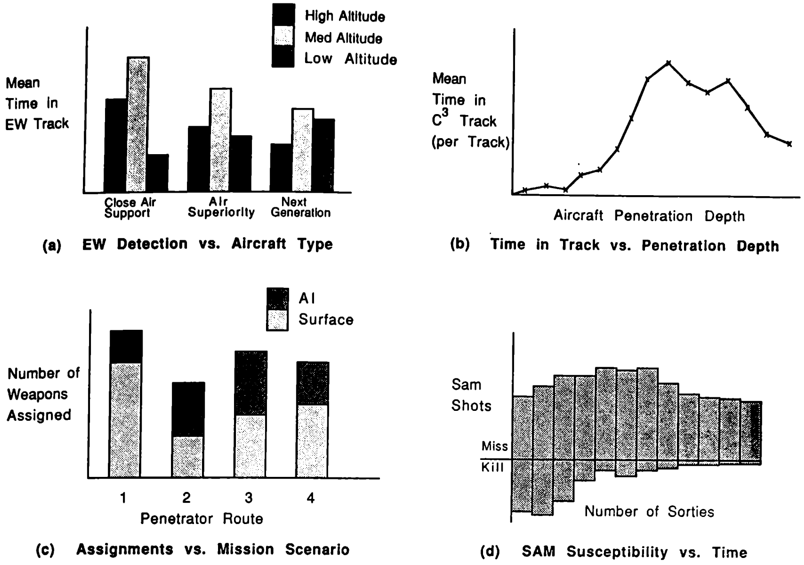
SAM FCR Aircraft Detection Histories
SAM FCR Track Establishment & Loss
SAM Launches
SAM Flyout Missile Histories
SAM Misses w/ miss distance & Pk
SAM Kills w/ miss distance & Pk

Figure 3 - EOB Recorded Data

Data is stored on magnetic disk in packed binary files during the real-time simulation runs, which are then post-processed to produce standard format magnetic tapes. These tapes are taken to other computer facilities where extensive data reduction software is used to analyze simulation engagement results.

Data analysis can focus on specific aspects of aircraft system performance against individual EOB models, or can incorporate data on aircraft interaction with several segments of EOB to evaluate aircraft performance against the entire ground defense network. Several examples of processed output from EOB data are shown in Figure 4.

Figures 4a and 4b are typical graphic representations of aircraft radar detectability fluctuations caused by several independent variables.



(a) EW Detection vs. Aircraft Type

(b) Time in Track vs. Penetration Depth

(c) Assignments vs. Mission Scenario

(d) SAM Susceptibility vs. Time

Figure 4 - Sample EOB Simulation Engagement Results

Typical data on C^3I performance against a given threat for several different mission scenarios is shown in Figure 4c. Figure 4d is an example of data analysis of the effect of pilot experience on SAM activity and effectiveness.

Initialization Software

The entire EOB model data base, with the exception of the SAM flyout and endgame parameters, is initialized through interactive menu-driven routines, allowing modification of any or all operating characteristics of EOB. As many complete threat laydowns as are needed can be stored on magnetic disk and called up between trials as needed, providing users the capability to instantly reconfigure the ground threat to suit their test scenario.

Large portions of the EOB model data base such as radar operating characteristics and aircraft radar signature patterns are contained in data files which are read into memory at simulation run time. The menu-driven initialization software saves only the file names rather than the entire contents of the file, breaking up the data base into more easily manipulated sections.

SAM model parameters are contained in separate data files from the rest of the EOB data base. Alteration of a data file is all that is required to change SAM operating characteristics. Multiple SAM data files may be stored on disk for easy access.

Since the EOB model data base is fairly large and complex, the initialization software contains extensive error checking functions. Errors and omissions, incorrect data formats, and table size overflows are all checked and flagged with messages to the user terminal and an error log file. These messages point to the exact nature and location of an error, simplifying the procedure for creating a new data base and reducing the time needed for users to become familiar with the data base. The menu driven routines also contain user help menus which can be accessed at any point in the initialization process.

Concluding Remarks

EOB is a data driven model. The source code is generic; specific radar, C^3I, and weapons systems are simulated by using appropriate input data. This allows completely different projects to tailor EOB's capabilities to their needs simply by building their own threat scenarios and saving them on disk. There is no need to modify source code, relink the executable programs, or reload the simulation in the real-time computer systems to change from one threat laydown to another or modify specific model operating variables.

Generic source code allows algorithmic modifications and upgrades to proceed in an unclassified environment, facilitating development and handling of the model itself and its related documentation. Configuration control is also simplified, since only a single copy of the code is needed to support multiple projects. Changes to the source code are naturally propagated to all projects using EOB.

The modular nature of EOB allows projects evaluating the performance of aircraft systems against specific aspects of EOB to build high-fidelity data bases for the portions of EOB they choose to emphasize, while using easily generated low-fidelity data packages for the non-essential models. This capability and the availability of both deterministic and probabilistic algorithms for radar detections and missile endgames are also useful when comparing offline simulation results (such as ESAMS, Tac Brawler, and Tac Suppressor) to those generated by EOB. Model flexibility allows users to tailor the operation of EOB to closely match that of other simulations.

List of Abbreviations

AI	Air Interceptor
AWACS	Airborne Warning and Control System
CAP	Combat Air Patrol
C^3I	Command, Control, Communications, and Intelligence
EOB	Electronic Order of Battle
EW	Early Warning
FCR	Fire Control Radar
FEBA	Forward Edge of Battle Area
GCI	Ground Controlled Intercept
HUD	Head Up Display
IADS	Integrated Air Defense System
ISSL	Integrated Systems Simulation Laboratory
MICS	Manned Interactive Control Station
MITL	Man in the Loop
Pd	Probability of Detection
Pk	Probability of Kill
RCS	Radar Cross Section
SAM	Surface to Air Missile

References

1. Luhn, J.M., "Tactical Air Combat in a Real-time Multiple-Engagement Simulation," AIAA paper 88-4601-CP, AIAA Flight Simulation Technologies Conference, Atlanta, Georgia September 7-9, 1988

89-3315-CP

REAL-TIME TACTICAL SIMULATION FOR WEAPON SYSTEM DEVELOPMENT

Dorsey B. Smith*
John J. Soderberg
Lockheed Aeronautical Systems Company
Burbank, California 91520

Abstract

The Weapon System Simulation Center (WSSC) at Lockheed has been developed to perform real-time simulation tasks: Aerodynamics, avionics, and pilot/vehicle interface for an aircraft under development, and the tactical supporting and threat environment in which target aircraft and their associated weapon systems must operate. To provide this capability WSSC has been organized into three tactical laboratories. Two of these are functionally identical projection domes, the third laboratory, currently under development, supports a motion base cockpit in which instantaneous maneuver loads can be generated and low-altitude turbulence simulated. Terrain and other visual features, including ground or sea targets, may be simulated in this system. Supporting facilities for WSSC include a software development laboratory and a hardware development laboratory/workshop. WSSC development in the immediate future addresses the completion of the motion base laboratory, preparation to support certain flight physiology and human engineering developments, addition of advanced auxiliary control stations, and provision of a display room for viewing of the full battle environment by groups of investigators.

Introduction

The Weapon Systems Simulation Center (WSSC) provides single-element flight simulations and interactive multiple engagement tactical mission simulations using two domed simulators. WSSC provides capabilities for the development and evaluation of:

- Crew systems, crew stations, and cockpit designs
- Pilot/vehicle interface
- Flight controls and flight handling qualities
- Aerodynamic designs and performance measurement
- Avionics system architecture, hardware, software
- Multiple engagement and cooperative tactics
- Aircraft low observability
- Air-to-air and air-to-ground weapon delivery
- Air-to-air and surface-to-air missile avoidance

WSSC is comprised of two fixed-base domed visual simulators, referred to as Tactical Mission Laboratories (TMLs) 1 and 2; a motion-base visual simulator is under development. Each domed simulator houses removable and interchangable cockpits; each has an independent mission control center, man-in-the-loop threat aircraft control, a ground-controlled intercept station, and facilities for real-time and post-mission performance and data review. The WSSC facility layout is illustrated in Figure 1.

Purpose. This paper describes the technical development process for WSSC, with emphasis on the design philosophy, the selected features, system functionality, and system capabilities.

Definitions. A few definitions are helpful to establish a common ground for this discussion:

Model. As used in this discussion, "model" implies a mathematical construct used to represent a tangible object. A model consists of mathematical equations, logical rules, and constraints. Again for purposes of this discussion, "model" also implies a computer program incorporating this construct.

Simulation. As used in this discussion, "simulation" means the process of exercising the model to determine the effect of changes in the constants in its equations, its logical rules, or its constraints.

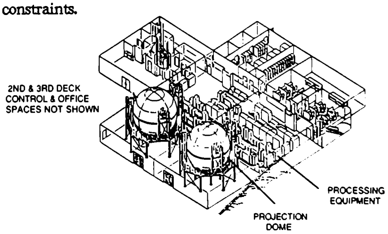


Figure 1. WSSC Facility Layout

* Member, AIAA

Real-Time. As used in this discussion, "real-time" implies simulation at an execution rate such that a wall clock in the modelled event space would run at the same speed as a wall clock in the real world.

Tactical Simulation. As used in this discussion, "tactical simulation" implies exercising models dealing with the conduct of air-air, air-surface, and surface-air warfare on the individual unit and the group-of-forces levels.

Weapon System Development. As used in this discussion, "weapon system development" implies the process of design and evaluation leading to definition of hardware and software for combat-capable avionic systems and air vehicles.

Why Simulation. Intrinsic complexity of modern weapon systems, rapid development of supporting technology, high cost of system development, and inherently long development lead times all point to the urgent need for information on which design may be reliably based. Computer simulation is a development tool with unique capacity both to develop and to evaluate design information. Simulation at the weapon system level has the capacity to create, in repeatable and quantifiable numerical form, the entire environment in which a weapon system must operate, a task which is essentially not "do-able" on a test range with actual hardware even if the luxuries of time and money were available.

Weapon System Design Issues. The designer of a weapon system would like to have a variety of information about a potential design, much of which is available through simulation:

- Sensor match to the signal environment
- Sensor match to weapon capabilities
- Sensor and weapon match to airframe capabilities
- Crew match to weapon system implementation
- Weapon system match to projected tactical environment

WSSC permits exercising each of these "matches" in a controlled environment - normally as a tactical engagement or series of engagements - and recording the results. Data may be collected, a single parameter changed, and data again collected to quantify the effect of the parameter change.

Unique Tactical Environments. Tactical engagements in the "real world" are essentially not repeatable, and in fact are not amenable to structuring in a useful way. By contrast, WSSC permits careful organization of a tactical engagement by the designer to best acquire needed design information, and then precise repetition of that engagement as many times as desired.

WSSC Design Approach

Design Tool. The WSSC design objective was to provide a full-mission flight simulator as a tool to enhance efficient systems design and development of high performance tactical aircraft. The system design concept was to include avionics, sensors, weapons, control and display systems, signature patterns, and vehicle performance. Flexibility was a key concept in WSSC design, in order that a variety of simulation customers may be served and so that new developments in weapons and tactics may be readily incorporated into the simulation.

Flexible Tactical Environment. The WSSC design was to permit the demonstration and evaluation of mission system effectiveness in a realistic tactical and threat environment prior to actual build and flight. In particular, effects on mission performance and vehicle survivability of known enemy threats were to be included. The phrase "Tactical Environment" is here used to mean the whole of the forces, both friendly and hostile, which the system design under test might be expected to encounter operationally. As simulated, these forces are equipped with accurate sensors and weapon performance as an essential element of the WSSC Tactical environment.

Flexible Tactical Platform. WSSC software models are developed to model accurately the physical systems being simulated. Wherever practical, WSSC software is modular and table driven. Accordingly, existing software models can be modified, or new models created, by modification or replacement of parameter tables or by mixing and matching software modules. WSSC has a library of software models which are available for use as is or which can be modified as required to support a new application.

Multi-Project. WSSC is designed to support a number of mutually-independent simulation projects simultaneously with a high level of physical security. Physical isolation and data independence between projects is a crucial element of the design.

Part-Task Simulation. The flexibility of WSSC software and hardware architecture provides the capability to do "part task" simulation tailored to customer requirements and budget. Part task studies can be tailored to address display, crew interface, or other investigations using WSSC cockpits with suitable controls, cockpit displays, HUD, and out-the-canopy visuals.

Aircraft Model

The WSSC high-fidelity aircraft model is a very complete model of a physical aircraft and its environment, as shown in Figure 2.

Aerodynamics. WSSC uses full fidelity six degree-of-freedom equations of motion which account for

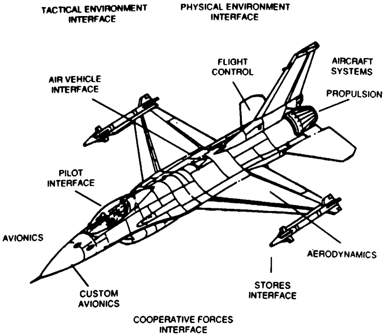


Figure 2. Aircraft Model

all inertial coupling terms, crosswind, turbulence, round earth, and include the effects of all primary and secondary control surfaces, flaps, speed brakes, nozzle deflections, chutes, and other force or moment producing systems. Array processors and function generation accelerators permit real-time use of large aerodynamic data bases.

Propulsion. WSSC uses data supplied by the customer or engine manufacturer to develop dynamic propulsion models of simple or moderate complexity. It is possible to integrate engine manufacturer's "state variable deck" into the simulation in real-time. When required, the design permits inclusion of engine/inlet installation effects.

Flight Control. WSSC can exactly simulate analog or digital flight control systems, sensors, and modes, including aerodynamic and structural interaction. This simulation is of sufficient fidelity to be utilized as a design and analysis tool to support the engineering development of flight control systems.

Avionics. Offensive and defensive avionics systems, sensors, mission processing systems, communication, navigation, identification, weapons management, and countermeasure systems can be simulated to a high, medium, or low degree of fidelity as warranted by customer requirements, or the simulation can be interfaced to actual avionic hardware running operational software. WSSC simulates a comprehensive mission avionics suite in order that the pilot may selectively detect, localize, and destroy or elude hostile threats in a realistic manner. The configuration of various avionics systems are based on customer specifications for the aircraft being simulated and the specific test to be performed.

Pilot Interface. The avionic system interacts with the pilot through cockpit controls and displays, accepting control inputs and parameters via switch actions, voice commands, and touch panel inputs. The actual crew interface of the aircraft under design/evaluation is emulated to the requisite degree of fidelity.

Cooperative Forces Interface. Interaction of cooperative forces with the simulated avionic system via various RF communication and navigation systems is also simulated; cooperative forces include other aircraft, ground controllers, and navigation aids.

Air Vehicle Interface. The simulated avionic system interacts with the simulated development air vehicle through simulated sensors and actuators. These provide the simulation with information about the state of the various systems and control surfaces.

Stores Interface. The avionic system monitors, conditions, and controls simulated stores. These interfaces provide both a means to control and a means to gather information from the simulated stores.

Physical Environment Interface. The simulated avionics system interacts with a simulated physical environment, which responds in accordance with standard geological and atmospheric physics, to prepare inertial reference data and air data. These data provide navigation and air vehicle status collection functions. The external environment also provides for diagnostics, maintenance crew interface, and stores inventory and health simulation.

Tactical Environment Interface. The avionics system interacts with the simulated tactical environment via simulated radio frequency and electro-optical sensors to gather information about objects in that environment which affect the mission of the aircraft. The "tactical environment" function performs sensing of the environment, generation of track files, and output of these track files to other system functions.

Custom Avionics. WSSC has the capacity to simulate new or innovative avionics designs, including advanced integrated avionics systems, offensive and defensive avionics systems, mission avionics, sensors, sensor managers, sensor fusion, weapons managers, etc. WSSC also has a library of accurately-modelled avionics simulations for specific U.S. and foreign aircraft, referred to as Mobile Platform Avionics. Figure 3 shows the design approach for Avionics.

Aircraft Systems. When required, vehicle management systems such as fuel management, hydraulic, landing gear, etc., are modelled and included in the simulation.

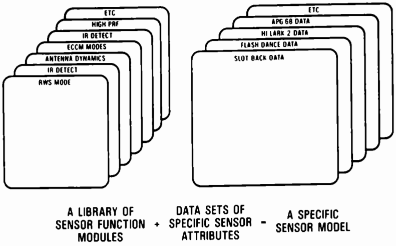


Figure 3. Sensor Design Approach

Tactical Environment Simulation

The essential elements of a tactical air warfare environment are simulated by WSSC, as shown in Figure 4.

Scale of Simulation. The WSSC tactical environment is predominantly an air-to-air and surface-to-air warfare environment. A total of 120 total airborne bodies may be simulated simultaneously; this total may include up to 38 aircraft, 30 missiles, 30 surface-to-air missile and/or EW radar sites, 20 expendables, and two GCI sites.

Types of Environments. The WSSC simulation deals with two different kinds of tactical environment: Beyond visual range and visual range.

Beyond Visual Range. Simulation of targets beyond visual detection range involves, in addition to maintaining position and position rate information on each target, maintenance of target aspect information for purposes of characterizing vulnerability to RF and IR sensors. The vulnerability of all targets must be assessed not only for the sensors of the aircraft under development/test, but also for cooperating forces such as GCI sites and other aircraft. If the target is radiating, the propagation characteristics of the emitter are also simulated.

Visual Range. Targets within visual range are principally engaged through visual reference from the cockpit, at ranges on the order of 5 miles and less. The visual environment is a demanding one to simulate; images must be accurate and correct to permit pilots to identify and classify, and must present the correct viewing aspect for the simulated relative motion. Further, such visually-valuable detection cues as sunlight glinting off canopies and wing and fuselage surfaces are extremely important for initial target detection. Similarly, the location of the sun and the characteristics of any weather phenomona tending to obstruct visual search must be taken into account.

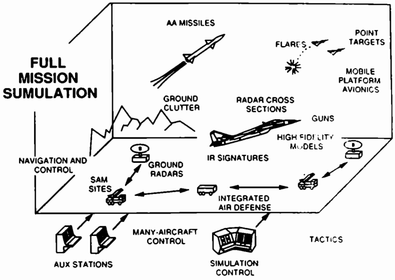


Figure 4. Tactical Environment

Hostile and Supporting Aircraft. There are two types of aircraft models: High fidelity and point target.

High Fidelity Aircraft Models. These are six-degree-of-freedom aircraft models having high fidelity aerodynamics, propulsion, and flight controls. Up to 16 high-fidelity models may be executed simultaneously. High fidelity models can be equipped with custom avionics or with WSSC Mobile Platform Avionics, and can be armed with guns and/or missiles. In addition to developmental aircraft, WSSC has currently modelled the F-15, SU-27, MiG-29, and Air Superiority Fighter (ASF) as high-fidelity aircraft, and all models are validated against customer data.

Point Target Aircraft Models. To provide additional background targets at little cost in computer resources, five- degree-of-freedom (i.e. no sideslip) point target models are available. An additional 22 point target models may be exercised simultaneously. Point target models have realistic performance, show up on other aircraft sensors, are equipped with basic avionics capability and weapons, and can shoot and be shot at.

Control of Aircraft. WSSC provides a variety of ways to control aircraft models, providing a flexible way to execute a variety of tactical scenarios.

WSSC Cockpits. Each of the WSSC simulation domes is equipped with a cockpit which provides realistic aircraft controls and displays and out-the-canopy visuals. High fidelity models, but not point target models, may be flown from these cockpits.

Auxiliary Control Stations. Each WSSC simulation dome can be supported by up to four auxiliary control stations, each of which may be used to fly any modelled aircraft. Each auxiliary control station consists of a throttle and stick with full HOTAS controls, a plasma panel with touch control to operate avionics and sensors, 16 master mode switches for rapid reconfiguration of sensors, avionics, and weapons, radar and RWR scopes, flight instrumentation, and a tactical display providing 360 degrees of coverage.

WSSC Brawler. WSSC Brawler is a real-time subset of the TAC Brawler tactical analysis program generated by DSA Incorporated. WSSC Brawler models the tactical decision-making process performed by a fighter pilot in both engagement and evasion of airborne targets. It is capable of executing both single-unit (one-on-one) and multi-plane tactics. From 4 to 7 total units may be controlled simultaneously; provisions in the system architecture provide for tripling this number with a modest hardware expansion. WSSC Brawler performs all tactical maneuvering within the performance constraints of the modelled aircraft, including power control, energy planning, and control of lift-management devices.

WSSC Pilot Model. The WSSC-developed pilot model can be used to fly any aircraft in the scenario. The pilot model can be scenario-driven or may follow commands from a GCI controller. The pilot model can be directed to fly waypoints, orbit, fly wingman or lead, attack, disengage, return to base, and a selection of other maneuvers. The pilot model can communicate with a GIB (guy-in-the-back) who operates avionics and weapons. Aircraft being flown from a cockpit, an auxiliary station, or WSSC Brawler may be reassigned in real-time during the simulation to a WSSC pilot model. Any aircraft model may be reassigned from a WSSC pilot model to an auxiliary control station or to WSSC Brawler; additionally, high fidelity models may be reassigned to a WSSC cockpit.

Weapons. The performance of the following tactical weapons is simulated:

Missiles. WSSC has an inventory of five-degree-of-freedom missile models (<u>i.e.</u> no roll), including the U.S. AMRAAM, Sidewinder, and Sparrow, and Soviet AA-6 to -11 air-to-air and SA-2 to -15 surface-to-air missiles. Up to 30 missile models may be exercised simultaneously. The missile aerodynamics, propulsion, autopilot, guidance, radar and/or IR seekers, and fire control radars are developed and validated against customer-provided off-line flyout models. Valid missile end game statistics have been generated using customer-supplied probability-of-kill tables that are a function of a specific missile against a specific target. Missiles may miss, kill, or partially damage a target based on these tables.

Guns. The ballistics, cyclic rate, and number of on-board rounds for modelled U.S. and Soviet aircraft is included in the simulation.

Electronic Warfare Environment.

Soviet Integrated Air Defense System. The model includes all existing and immediately-projected SAM types, together with their respective fire control radars, long range acquisition radars, and height finders. The model permits either centralized, autonomous, or self-protect command and control. Up to 400 EW/GCI or SAM-firing units may be pre-positioned in geographic coordinates; of these, up to 30 may be dynamically selected at any one time to engage up to 16 targets.

Expendables. The effect of flares is modelled for real-time simulation against IR missile seekers.

Electronic Countermeasures. Both U.S. and foreign electronic jammers are modelled for real-time simulation. Missile seekers, launch platform radars and SAM fire control units react to specific jammer types based on their specific ECCM capabilities.

Clutter. Radar missile seeker and launch platform radar detection and tracking performance is modelled to include main-lobe and side-lobe ground clutter.

Signatures. Radar cross sections are modelled for all aircraft and missile types, for multiple frequency bands, as a function of aspect. IR signatures are modelled for all high fidelity aircraft models; a generic IR signature is provided for point target models. Flare and missile IR signatures are also generically modelled.

IR Environment. All WSSC aircraft models include aspect- and power-dependent modelling of infrared emission. Sensor detection models include reaction to ground-generated interference, sun position, and cloud attenuation. Performance of IR-guided weapons includes the effect of IR countermeasures.

Tactical Cockpit Simulation

Each of the two simulation domes is equipped with a complete cockpit from which the aircraft under development/test is flown. WSSC cockpits may be custom-made with a particular aircraft or crew interface in mind, or a "generic" cockpit maybe used when the cockpit configuration is not one of the system design elements under test. The available controls and displays include head-down multifunction displays, head-up displays, helmet mounted displays, voice interaction, hands-on throttle and stick controls, and dedicated switches and indicators. A cockpit is shown in Figure 5.

Controls. Control arrangement for flight safety and for combat efficiency is a design element which is particularly amenable to optimization through WSSC simulation.

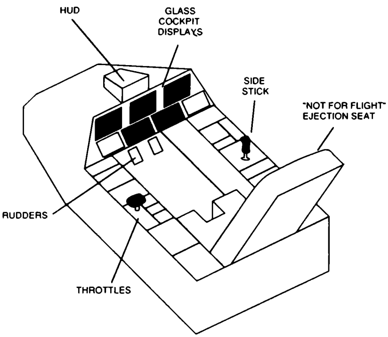


Figure 5. Dome Cockpit

Stick, Throttle, and Rudders. Center or side-arm stick arrangements may be used. The control may be stationary (pressure sensing) or moving; feedback may be mechanically or hydraulically simulated. The WSSC system is particularly well adapted to evaluation of fly-by-wire system characteristics and arrangements. Throttle arrangement, either single or multiple lever, is modifiable throughout a wide range of physical arrangement. Electrical switches are included on the throttles as specified, and the position of these is modifiable. Standard rudder pedals are provided, with adjustable distance and throw.

HOTAS. The "Hands on Throttle and Stick" (HOTAS) approach to cockpit design is fully implementable in WSSC cockpits.

Weapon Control. A variety of cockpit arrangements are available for weapon control; normally, weapon controls are designed to be interactive with the vehicle weapon management system. Arrangements for this interaction may be evaluated, as may the arrangement of switches and indicators.

Displays. The display of flight, sensor, tactical, navigation, and vehicle system information is becoming one of the most critical elements of weapon system avionics design. WSSC has been designed to contribute strongly to display development work through testing in realistically-simulated flight and tactical flight environments.

Glass Cockpit Graphics. WSSC cockpits are normally configured with multi-function CRT-based color displays; the WSSC Graphics group has the in-house capacity to emulate any desired display organization and present any desired display graphics. Current cockpit configuration is for color CRT displays; this varies from aircraft to aircraft.

HUD. The WSSC cockpits are normally configured with a Heads-Up Display (HUD). The simulation provides the necessary aircraft performance and attitude data, and the display may either be emulated on a CRT or derived from the actual HUD hardware.

HMD. WSSC has the advanced prototype Helmet- Mounted Display (HMD) technology presently under evaluation.

Aural Cues. WSSC cockpits make use of extensive aural cue simulation to provide additional realism to the flight and tactical simulation.

Reconfigurable. It has been an essential WSSC design criterion that the cockpits be reconfigurable, both to permit conformance to design changes within a given project, and to permit accommodation of a variety of projects. A subsequent section will discuss the development shops used to perform the reconfiguration work.

Projection Dome

Visual Information. The dome system produces the visual effect of "looking out the canopy"; terrain, cultural features like roads, cities, and airports, weather and clouds, and other aircraft are all shown with appropriate aspect and motion.

Terrain. The landscape visuals are projected to appear as they would from the air vehicle's altitude, from operational ceiling to "on the deck". Visuals are of two types, generic or location-specific. Generic flat terrain with agricultural patterns is shown as a default when terrain features are not a consideration in the tactical problem. Representation of actual terrain, down to individual rocks and bushes if warranted, may be constructed from charts, imagery, and/or numerical databases. Sun or moon position and shadows are correct for mission time-of-day and time-of-year. Ocean visuals may also be simulated, including wind-dependent sea state.

Cultural Features. Roads, buildings, canals, and any other man-made feature may be created generically or may be made to duplicate actual features. Objects on the ground, or on the ocean surface, may be reproduced with any desired degree of fidelity.

Weather. Clouds, fog/haze, and precipitation may be introduced, with complete or partial/random obscuration of terrain and cultural features.

Aircraft and Other Objects. There are currently 63 aircraft types in the visuals library. The average resolution for aircraft is 200 to 300 faces, with up to 800 faces in use. In any case, sufficient information is displayed to permit visual aircraft identification within appropriate range limitations. There are currently over 120 models of various types in the database, including land vehicles, ships, missiles, radar sites, tanks, and missile carriers. Weapons in flight are also shown.

Visual Background Projection System. The visual projection system consists of a dome, background projectors, target image generators, and a host processor. A dome projection system is shown in Figure 6.

Dome. Each projection dome is 28 feet in diameter and is constructed of externally-cantilevered aluminum, with internal sphericity accurate to +/- 1/4 inch. The pilot's eye is positioned at the center of the dome. The interior of the dome is spray-coated with a special projection screen paint, providing a nominal reflectivity index of 2.

Projection System. The projected image is generated by a G.E. COMPU-SCENE IV computer image generation system. The image as seen from the cockpit encompasses +/- 180 degrees in azimuth and + 90 - 45 degrees in elevation as seen by the pilot. The system utilizes 4 G.E. PJ5155 Talaria projectors, image-aligned to within approximately 10 arc-minutes in azimuth and elevation. Overall image resolution is 8 - 11 arc-minutes. Image interlaced field refresh rate is 60 Hz. Image brightness under simulated daylight conditions is greater than 2 foot-lamberts. Functionally, each Compu-Scene IV supports 64 moving models.

Host Processor. The projection system is hosted by a modified Gould 32/9780 dual CP processor, which receives information for image location, kind, and dynamics from the associated processing node via a Gould 3050 shared memory system. The host is coupled to the special-purpose computer and video stage of the Compu-Scene IV. A block diagram of the processing equipment is shown in Figure 7.

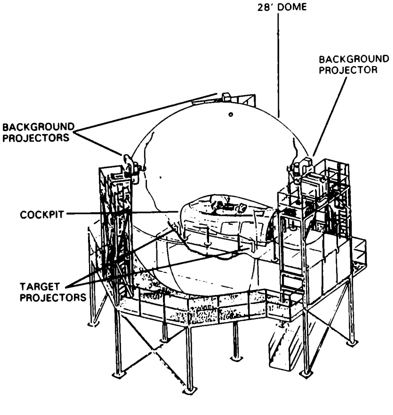


Figure 6. Projection Dome

Target Projection System. High-resolution targets are projected by 4 servo-driven panoramic projection heads, whose location is also shown in Figure 6. The target projectors permit a minimum of 3:1 improvement in projected resolution over the background projectors. Up to two targets may be projected at a resolution of less than 1 arc-minute in any dome location; a bright image is superimposed over the image projected by the background system. Up to Mach 2 relative aircraft speed and 250 degrees per second roll rate are displayable.

The target projection systems direct projection of high-resolution target images onto the dome projection surface in such a fashion as to accurately and realistically depict the targets as they appear from the pilot/crew viewpoint. for the Dome 1 implementation, the target projectors are capable of providing up to two separate targets. For the Dome 2 implementation, the target projectors are capable of providing up to four separate targets.

The target projectors are capable of projecting these images on any portion of the dome surface which is visible to the pilot.

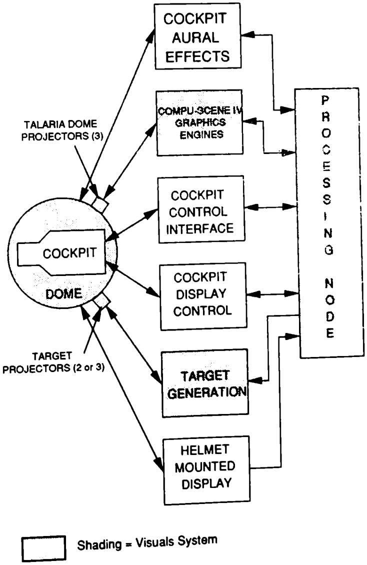


Figure 7. Visuals Processing

A dynamic target allocation routine determines target priorities for assignment to the target generation channel based on scenario-driven parameters. The number of assignable targets is limited by CIG/target projector constraints. The remaining targets not assigned to target projectors, up to thirteen, are placed in the background scene.

Aural System

The WSSC Aural System provides three-dimensional digital sound location technologies for simulation use in assessing pilot/crew reactions to aural stimuli.

The Aural system consists of the following subsystems:

- Environmental Sound Generation System (ESGS)
- Intercommunications System (ICS)
- Voice Recognition System (VRS)
- Voice Synthesis System (VSS)
- Aural Data Analysis System (ADAS)
- Voice Disguising System (VDS)

Aircraft Cues. The ESGS provides a full range of environmental sounds which are discernible to the pilot/crew in the performance of their mission during a simulation execution. The ESGS reproduces all interior and exterior sounds required during a simulated mission. For example, by referencing libraries, sounds such as in-air compressor whine, jet blast, missile launch, ground activity such as runway crunch, tire skids, engine start, and weather such as thunder, lightning strike, rain, effects are reproduced. Environmental sounds for which the position of the sound source is normally apparent to the pilot during actual flight conditions is represented directionally in the headset presentation.

Communications. Simulates the vehicle interphones and radio systems, provides the intercom for lab facility

Voice Recognition. Permits pilot/crew to interact verbally with aircraft controls and functions

Voice Synthesis. Provides aural cueing to the pilot/crew for state conditions of the vehicle

Aural Data Analysis. Provides for digital recording and playback and creation and modification of the ESGS sound libraries

Voice Disguising. Provides a real-time means of altering a speaker's voice patterns

Auxiliary Control Stations

Each WSSC simulation laboratory is equipped with three auxiliary control stations of two types: "Aux Stations" and "C-CUBED Stations".

Aux Stations. Two Aux Stations are provided per mission laboratory. From an Aux Station a WSSC operator can designate any aircraft in the simulation, either a five- or a six-degree-of-freedom model, assume control of it, and fly it as a participant in the tactical simulation. An Aux Station is shown in Figure 8.

Aux Station Displays. The display screen is divided into two parts, tactical/flight and avionics. Flight display, in a format similar to a HUD, provides for attitude orientation, IAS and Mach number, heading, steering command, altitude, and altitude rate. A tactical situation indication with either a 28- or a 360-degree degree field of view is included, together with a gunsight and missile engagement display. The lower segment of the screen presents sensor displays for the Mobile Platform Avionics suite discussed in an earlier section. It includes radar (RWR) and IR warning displays. It also includes weapon system status displays sufficient for operation of the simulated aircraft model.

Aux Station Controls. The principal controls of the Aux Station are the throttle and sidearm-mounted stick; these are used for dynamic control of the aircraft model in a manner analogous to flying an airplane. The stick and throttle are equipped with "Hands on throttle and stick" (HOTAS) switches and activators to control weapons, sensors, communications, and speed brakes. The other "non-aircraft" controls allow the operator to manage the interaction of the Aux Station with the simulation.

C-CUBED Stations. One C-CUBED Station is provided per mission laboratory. From a C-CUBED Station a WSSC operator can establish voice communication with any aircraft in the simulation

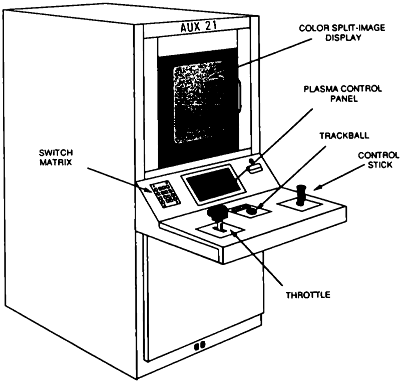


Figure 8. Auxiliary Control Station

in order to provide ground controlled intercept (GCI) tactical vectoring information.

C-CUBED Station Displays. The display is divided into two segments, as for Aux Control. The top segment represents a top-down view of the selected battle area, within which a center screen marker represents the center of the battle area and two range rings represent the selected display range and half the displayed range.

C-CUBED Station Controls. The principal controls of the C-CUBED station are the trackball and the co-located hook and cancel switches.

System Interfaces. The Auxiliary Control Stations have their data and control source in the real-time processing node machine designated AUX/MPA, the "MPA" being an acronym for Mobile Platform Avionics as discussed in Tactical Simulation section of this paper.

Simulation Control

Control of the execution of the WSSC simulation involves three general task categories: Operation of the processing system, operation of the tactical scenario, and operation of data extraction.

Processing Operation. Control over the real-time processing node is exercised through the real-time executive software using the control console.

Real-Time Executives. The WSSC simulation is exercised through the processing node machine real-time executives, two per dual-processor machine. The executives call tasks, handle interrupts, perform external communications, and all other roles associated with an executive program. The first executive on the processing node I/O machine, designated E0, is the master executive for the simulation.

Control Console. A control console is located in the operations room of each Tactical Mission Laboratory. Each console consists of a set of color CRTs repeating the dome cockpit displays including the cockpit HUD, an intercom and control screen for voice communication with the rest of the system, and two DEC VT-320 terminals for communicating with the I/O machine in the processing node.

Control Menu. The E0 executive displays a menu at the control console which permits selection of operating modes: Run, Freeze, Initialize, Terminate. These mode selections are duplicated with pushbuttons in the dome cockpit, but without the associated displays. The console menu displays the run number, the frame count, the status of the other executives in the simulation, the current mode requested, and error states if any. Other menu pages are also available for system timing and debugging processes. The menu also permits selection of the scenario number to be run.

Scenario Operation. Tactical simulations are organized on the basis of the "scenario", an initial configuration of all the participants in an engagement, including types, numbers, and locations of aircraft, weapons, air defense sites, the time of day, weather, and any other information of significance. Being able to reset to a given scenario is one of the guarantees of repeatability ir the WSSC simulation. The progress of the execution of the scenario is tracked by the system operators using the CRT displays of Simulation Information Management System (SIMS) equipment, also positioned in the control room, and the Auxiliary Control Station C-CUBED functions.

Data Extraction Operation. The analytical purpose of WSSC dictates the collection and analysis of information collected during the running of the scenarios. This information is obtained via the "data extract" process, which is menu-selected from the console.

Processing Node

Gould Real-Time Processors. The WSSC simulation for each dome is driven by a processing node of real-time machines: Eight Gould Concept/32 9780 and one 6745 SelPAC. These run under the Gould MPX-32 real-time operating system. These are dual-processor machines; each processor is equipped with a floating-point accelerator and a multiply accelerator. Each of the eight 9780 machines is equipped with 128 KBytes of Cache memory, 12 MBytes of private memory, and 4 MBytes of shared memory. The 6745 is not on shared memory, and has 16 Mb of private memory. As discussed below, selected machines are supported by Floating Point Systems 5300 array processors or by RDH Simulation Function Generation Accelerators. A processing node block diagram is shown in Figure 9.

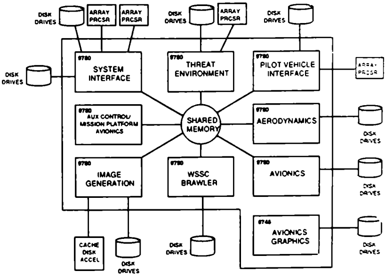


Figure 9. Processing Node

Functional Task Distribution. The processing node machines are functionally dedicated to the tasks described below:

- Aerodynamics (AERO) - provides performance calculation for each active body in the simulation; supported for processing by a Function Generation Accelerator
- Avionics (AVI) - provides sensor information for each active body in the simulation
- Pilot/Vehicle Interface (PVI) - provides for control of active bodies, their avionics, and their vehicle systems. Hosted by the Avionics machine via a Gould Inter-Computer Bus Link (IBL) vice shared memory.
- Threats/Weapons (THR/WEP) - provides for simulation of the battle environment, including radar sites and missiles, supported by an FPS 5300 array processor
- WSSC Brawler/Avionics 2 (BLR/AVI2) - supports the WSSC Brawler tactical model, and provides off-load support for the Avionics processor
- Input/Output (I/O) - provides relative motion computation for all active bodies, supported by two FPS 5300 array processors, and provides for control of the simulation
- Computer Image Generation (CIG) - provides for control of and image generation for the the dome background and target projectors, database I/O supported by a cache disk accelerator
- Auxiliary Control/Mobile Platform Avionics (AUX/MPA) - provides for control of the two auxiliary control stations
- Data Extract (DAT EXT) - performs data extraction and hosts the Logicon SIMS processor subsequently described

Shared Data. The functional task divisions described above require the sharing of some common data between processors. Shared data is in the form of variables used by more than one machine of the processing node; these are resident in the node's shared memory, are named in common throughout the system, and are used/updated through use of a software semaphore to ensure currency and validity.

I/O Provisions. External interfaces by each processor of the node to other node processors are accomplished through shared memory as discussed above. Interfaces to other non-node resources are via the Gould SELbus and High Speed Data (HSD) interface card. The generalized processing scheme is shown in Figure 10.

Performance Considerations. The principal concern in operation of a real-time system is frame-time. A "frame" is the span of time during which the processing must be accomplished. In WSSC's case, the span of time for processing is determined by the need to refresh the background image projected on the dome. The image is interlaced at a 60-Hz rate, which essentially means that a substantial portion of the entire simulation computation must be performed at a 30-Hz rate, or every 333... milliseconds. There are background tasks which must also be accomplished by the system processors, in addition to the real-time computations, so the available frame time is on the order of 300 milliseconds. Task rates vary from 5 to 120 Hz depending on the task; most are 30 Hz or higher. Figure 11 illustrates a typical allocation of processor tasks to the available frame time, and provides an indication of the current status of frame time utilization by processor as a percentage of the total frame.

There is approximately a 15 percent "buffer" of processing time, on the order of 5 MIPS, during nominal average load operations. The system design allows single-frame overruns, providing the system can "catch up" on the next frame.

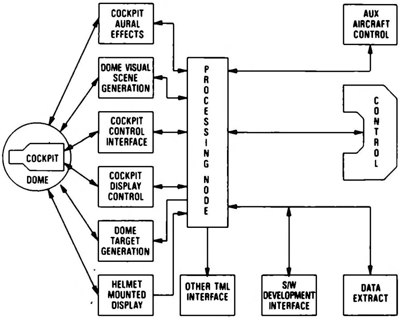


Figure 10. Processing Organization

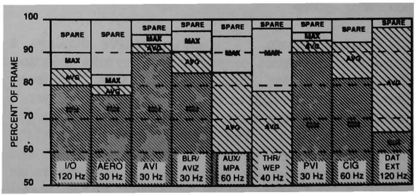


Figure 11. Frame Time Utilization

Data Review and Reduction

The purpose of WSSC simulation is to collect data in support of system design. Data collection requires decision-making about the kind and quantity of data to collect. This is done in three ways: Concurrent review is necessary to ensure that the right thing is being simulated as the simulation progresses, a "Quick Look" review is performed immediately after a simulation run to give a fast look at the event, and a variety of detailed analyses are subsequently performed on the data to extract all the desired information.

Concurrent Review. WSSC makes use of a Logicon Simulation Information Management System (SIMS) to permit examination of the progress of the simulation concurrently with its operation. SIMS provides several operator-selectable displays for immediate observation and monitoring. Some of these are simulated three-dimensional displays, which allow the user to interactively rotate the azimuth and elevation of the ground plane to the desired viewing angle, and which have zoom view capability:

- Situation
- Integrated Air Defense System
- Navigation
- Cockpit View

Other useful displays include:

- Ground Controlled Approach (GCA)
- Strip Chart
- Continuous Parametric Data (CPD)
- Event Data display

Quick Look Review. A SIMS post-simulation "quick look" review provides all displays available in the real-time environment, and adds the following:

- The Missile Engagement Profile
- All Aspect Maneuvering Index
- Pilot Performance
- Energy Maneuverability
- Energy State Time History

Detailed Review. WSSC incorporates data extraction at very high data rates on a large number of parameters during real-time operation. These data are captured real-time and extracted to disk by each of the several computing-node machines. At the end of each scenario execution, these machines are downloaded onto a single storage medium, normally one or more 300-Mbyte disk packs. This voluminous and permanent set of test data is then available for detailed analysis.

Supporting Shops and Laboratories

Hardware Support. WSSC requires specialized hardware support in several crucial areas: Cockpit make-up, special-purpose hardware fabrication, and system maintenance.

Graphics/Visuals Development Laboratories. "Graphics" deal with cockpit and auxiliary control system displays, whilst "Visuals" deal with the scene as viewed "through the canopy".

Graphics Development System. The graphics development system includes two Evans and Sutherland PS-350 graphics workstations hosted by a MicroVAX II processor, and two G. E. Graphicon workstations hosted by a second MicroVAX II processor. The MicroVAXes communicate with each other using DECnet, and with the Gould 9780 Pilot/Vehicle Interface host using Ethernet.

Visuals Development System. The visuals development system includes two Tektronics 4125 graphics workstations directly hosted by a Gould 9780 processor. The Visuals databases are stored on 300 Mbyte disk drives; database access is supported by a Gould cache disk accelerator. The visuals development system also includes five Apollo workstations, one of which hosts a digitizer for converting physical models to visual images. The Apollo workstations communicate files with the Gould 9780 visuals host machine using Ethernet.

Software Development Laboratory. WSSC is continuously involved in the process of software development.

- Unit level coding and test
- Functional integration testing
- System integration testing

The software development laboratory must necessarily support the process both with processing hardware and with simulation hardware. Concurrently, actual simulation system operations must be permitted to go on without interference. Accordingly, substantial hardware resources are needed. The software development laboratory consists of ten Gould Concept/32 9780 and 6745 processors, two Gould shared memories, a DEC VAX 6310, and simulation hardware. A block diagram is shown in Figure 12.

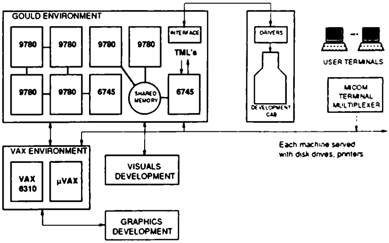


Figure 12. Software Development Laboratory.

Future Development

WSSC is now close to reaching the point in its development cycle at which a firm operational baseline capability is established. The next steps will require expansion of operating capability into desirable but less immediately urgent areas.

Motion Base. A motion base cockpit has been procured.

Radar Ground Mapping. The Digital Radar Land Mass System (DRLMS) has the capacity to generate terrain-oriented data for use in the simulation of terrain-following radars, ground-mapping radars, and similar devices. DRLMS has been procured.

Ground/Sea Target Attack. Motion base cockpit and DRLMS combine to give a capability for simulating ground attack missions. Instantaneous acceleration cues are important for this kind of simulation, particularly so to the evaluation of cockpit instrumentation and controls under conditions of high-frequency turbulence encountered in these types of tactical operations.

Carrier Operations. The motion base cockpit, in combination with WSSC's accurate simulation of air vehicle aerodynamics, control responses, and avionics, and realistic visuals, can permit accurate pre-build assessment of elements of the carrier suitability of potential carrier-based aircraft designs.

Physiologic Data Collection. The WSSC tactical simulation will provide an opportunity to collect a variety of physiologic data under realistic simulated combat conditions.

Video-Based Motion Studies. WSSC simulation will provide Human Factors engineers with the ability to video-tape pilot motions in the context of tactical maneuvering task performance.

Viewing Room. Currently, full-fidelity visuals are available only from the pilot's seat in the cockpit. A higher-fidelity projection-based screen with sufficient size and brightness for display in a small room would enhance several possible WSSC applications.

Conclusion

WSSC is an Engineering Development tool for the conduct of system design and development for avionics, flight controls, man-machine interface, avionics systems, and many other specialties of vital interest to designers of tactical aircraft. As a real-time simulation facility with a broad-based capability to simulate the air warfare tactical environment, WSSC has substantial potential for reducing technical risk, schedule risk, and cost throughout the system development cycle.

Acknowledgements

The authors would like to express their appreciation to the managers and staff members of the WSSC organization for their assistance and cooperation in preparing this paper. In particular, we would like to thank Mr. M. E. Ford of the WSSC Simulation Systems Design Department for collecting the necessary technical information and for his assistance with with preparation of the manuscript. More fundamentally, we would like to acknowledge our debt to the many people whose dedication, competence, and hard work have made WSSC a reality.

89-3338-CP

COOPERATIVE SIMULATION EFFECTIVENESS ANALYSIS

Paul A. Marchisotto
David A. Blancett

19 June 1989

Northrop Corporation
Aircraft Division
One Northrop Avenue
Hawthorne, CA 90250

Introduction

Classically, the requirements definition and systems analysis needed to support the design of tactical aircraft have been based on present perceptions and historical extrapolations about the nature of future combat tactics. No means of verification as to how combat could change based on new technologies were available. Changes in combat tactics due to changes in system capabilities must be recognized before they occur; the requirements definition and systems analysis process must be adjusted accordingly in order to avoid developing ineffective designs. Manned simulation provides a means for exploring new tactics and reducing the risk associated with aircraft design.

The requirements definition and systems analysis work associated with optimizing the design of tactical aircraft has taken the form of timeline and simple engagement analysis using unmanned digital simulation models. Though these methods are useful for defining basic requirements and rank ordering both alternate technologies and aircraft concepts, they are limited in their operational realism. "Real world" environmental elements and human decision making can cause the performance of a weapon system to be quite different than predicted by unmanned digital simulations. An example might be sensor coverage requirements based on unmanned simulations of assumed one-on-one engagement maneuvers. In a more chaotic environment, which considers pilot creativity and decision making ability, the probability or frequency of occurrence of such maneuvers may indicate that totally different coverages are desired. If early system requirements and design decisions were based on the unmanned simulation results alone, reduced mission effectiveness or an unjustified increase in cost could result.

Higher fidelity, unmanned simulation models, such as AASPEM and Tac Brawler, can be introduced to evaluate concepts in a more "realistic" environment. Much insight can be gained from this work; however, the "hands-on" experience, which would permit true pilot experimentation with new technologies and aircraft designs, would still be lacking.

In manned simulation, pilots are able to obtain hands-on experience with potential future designs and technologies in a projected combat environment. Previously, pilots have assumed the role of consultant or advisor in the requirements definition and system analysis process, and as direct participants only after the aircraft had been built. It became clear that in order to predict the technology-driven changes in combat tactics and adjust the design process before the aircraft had been built, pilots needed to become participants in the process and not merely consultants.

The development of multi-player manned simulation facilities has provided a major step toward realistic requirements analysis in support of tactical aircraft design. The use of manned simulation also has the benefit of feeding tactical "lessons learned" to the unmanned simulation models and provides a benchmark for the pilot behavior rules which govern models such as Tac Brawler and AASPEM. The unmanned simulation models can then be used to assess broader excursions of more system parameters. It also enhances concept traceability between design changes, beyond that achievable by manned simulations alone.

A multi-layered, interactive analysis approach has been developed and successfully applied by Northrop, as depicted in many-vs-many (m-v-n) unmanned models, the high fidelity unmanned models, and the manned simulation. This coordinated application of both manned and unmanned simulations to analyze advanced aircraft design concepts and technologies is called Cooperative Simulation Effectiveness Analysis (CSEA). Using the CSEA approach, increased "real world" impacts have been introduced throughout the requirements analysis and aircraft design process.

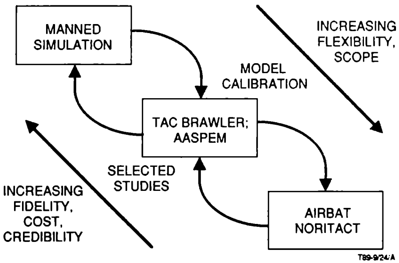


Fig. 1. Cooperative Simulation Effectiveness Analysis (CSEA) Model Hierarchy

Development History

1-v-1 Unmanned Simulation

Operational inputs from experienced pilots have always been an element of the requirements analysis process. However, in most instances these have taken the form of discussions between pilots and analysts, followed by the analysts performing the technology trade studies independently. The first attempt to bring pilots directly into the system analysis/requirements definition loop was through a modification of the Tactics 1-v-1 air combat engagement model. Northrop analysts converted Tactics from a batch program into an interactive tool called NORITACT, for Northrop Interactive Tactics, which could be modified by the user as the engagement progressed to reflect various tactical employment options.

By using NORITACT cooperatively analysts and pilots could both benefit from the insight each other provided into the problem solution, with NORITACT acting as the conduit. The close association of analyst and pilot provided an effective means for each to explore the needs of the other. The pilot was exposed to the system integration problems and operational employment challenges posed by advanced technologies. The analyst gained a better understanding of current tactical operational doctrine, pilot response to combat situations, and employment options for advanced technologies.

Although a highly useful tool, NORITACT is lacking in two respects. First, it is not a real-time interaction and, therefore, allows the pilots time to think through their tactical options prior to selection, a luxury not often afforded to pilots in actual combat. Second, NORITACT, as a 1-v-1 air combat model, does not provide understanding of a multi-aircraft combat environment.

M-v-N Unmanned Simulation

Various models exist which address m-on-n combat with various levels of fidelity. The major challenge in each of these m-v-n air combat models is to control the aircraft in a realistic operational manner. Essentially, the pilot information gathering and decision processes must be modeled with enough flexibility and alternate actions to allow it to be applied to advanced technology aircraft design studies. Without a means to introduce realistic pilot behavior, the credibility of the results will suffer. Several m-v-n air combat models were investigated for application to the requirements definition process with various degrees of success. The Tac Brawler code was adopted as the principal tool due to the robust yet flexible manner in which it models pilot behavior.

Pilot interface with Tac Brawler is accomplished through production rules and the adjustment of weighting factors associated with the value decision logic. This interface provides good flexibility for implementing pilot behavior associated with new technologies, as well as the ability to control, through the production rules, a range of pilot actions.

Based on the tactical mission scenario, mission objectives, aircraft design concept, and associated technologies, the pilot community defines how the aircraft will be employed, which is then translated into Tac Brawler production rules. Following the initial definition of employment tactics, the pilot reviews the resulting unmanned simulations to assure the desired tactics are occurring and that they truly represent an efficient use of the aircraft. This pilot review of the unmanned simulations represents a learning curve, not unlike that associated with the introduction of a new aircraft into operational service. The primary weakness in unmanned simulation is its lack of both a true pilot-vehicle interface and the dynamic decision process which would occur in the real world. The pilot is denied the opportunity to truly interact with the aircraft against the threat in a time-critical combat situation.

Early Manned Simulation

Development of a manned simulator to permit multiple interactive players and simulation of complex scenarios extended analysis to a level which overcame these limitations. This, coupled with NORITACT's success in introducing the pilot into the analytical requirements definition process, resulted in Northrop's 1982 decision to build a multiple-engagement, tactical simulation facility that would allow pilots to directly participate in the concept development phase of tactical aircraft design.

Prior to 1982 Northrop had been using two dome simulators, which could not communicate with each other and were used primarily for cockpit design, pilot training, flight control development, and hardware and software integration, all of which are classical simulator applications. The decision to also apply these simulators to concept development studies dictated some major design changes. To create an environment in which advanced air combat could be investigated, a greater number of real-time interactive participants was needed.

Predictions of future force engagements indicated a need to simulate at least three times as many threat (Red) aircraft as friendly (Blue). This requirement caused the new simulator facility to increase simulation participant capacity by the addition of pilot interactive stations. The unique contributions that an actual pilot makes to the analysis process showed that at any participation level the pilots' insight was valuable. If computer controlled participants had been used instead of actual pilots, the original purpose of introducing operational experience and creativity into the design process would have been diluted. This applied to Red participants as well as Blue. Red counter-tactics are equally important in determining the effectiveness of advanced aircraft concepts and alternate technologies. If an advanced technology can be easily defeated by a creative Red player, its utility is suspect and as such should be reconsidered.

Northrop decided to increase its simulation capability for concept development applications by linking the two existing dome simulators with two Manned Interactive Control Stations (MICSs). Each MICS was composed of a 19-inch monochrome monitor which functioned as the aircraft displays and a simplified stick and throttle. All vehicle control was through the stick and throttle and all aircraft subsystem control was through touchscreens attached to the surface of the 19-inch monitor. This provided four simulation participants, subsequently divided into one Blue participant and three Red participants, to conform to the projected threat force ratio advantage.

In using this configuration for numerous concept development studies in 1982, many insights were gained regarding the nature of operations in the BVR combat arena and the operational utility of the AMRAAM missile on various weapon system platforms. One of the tactical lessons learned very early in the application of this facility was that a single ship Blue force (1-v-3), which obviously precluded cooperative tactics, had difficulty exploiting some technologies to the fullest. Further, it was difficult to compare the results from the manned simulator with the unmanned simulation results since the latter models were operating almost exclusively with cooperative two-ship Blue elements and performing much better than the single-ship participant in the manned simulator.

Simulation Cross Validation

Two more MICSs were added to the simulator facility in 1983. This permitted further investigation into certain technologies and weapon systems concepts which had proven to be very effective in the simulations and showed promise from a technology maturity standpoint as well. Further, it was hoped that by introducing a cooperative employment capability for Blue force (2-v-4), the level of cross validation of manned simulation results with unmanned simulation results would be improved.

After one application of the new six-participant simulation capability, correlation of effectiveness trends between manned and unmanned simulation results was achieved. Correlation was not achieved, however, for the absolute magnitudes of the effectiveness results. This highlighted several weaknesses of the cooperative simulation approach. First and foremost of these was the ability to achieve statistical validity in the data. Because air combat is such a dynamic environment, the results are highly variable from engagement to engagement. This requires that many manned and unmanned simulated engagements be run in order to establish the proper trend, or cause-and-effect relationship, driven by the introduction of a new technology or tactic. While it is standard procedure to replicate the unmanned simulation models to the extent necessary to achieve statistical validity, the amount of time needed to do so in the manned simulation has to be allowed for.

There is a difference in variability of effectiveness results between manned and unmanned simulations. This difference is due to pilot experimentation and random error in the manned simulations. For instance, real pilots may not have sensors in the proper mode and intercepts will be missed. On the other hand, unmanned simulation "pilots" do not make pilot-vehicle interface errors, tend to perform with less random error, and have no prior knowledge of mission conditions unless they are programmed in a priori. Any comparison between manned and unmanned simulation must account for the variability of individual pilot capabilities and a degree of random error in their performance. Unmanned simulations represent one point on a pilot's learning curve; that is, a consistently performing pilot maintaining a constant ability. The pilot in a manned simulation has a greater variability of results and therefore requires even more simulated engagements in order to establish statistical validity.

Another weakness that became obvious in this approach was that, due to differences in the software, certain key environmental aspects and subsystems were not behaving equivalently between the two simulations, thereby generating differences in the final mission effectiveness results. A means was needed to assure consistency in approach, assumptions, procedures, and modeling behaviors, thus allowing cross validation of results at all levels. To achieve this, a formal interface group was established and its role in CSEA is diagrammed in Figure 2.

This group, which meets on a regular basis, has concerned itself with common threat data; mission assumptions, scenarios and objectives; Rules of Engagement (ROEs); subsystem model behavior; and analytical techniques. Subsystem and environmental models, such as the sensor performance algorithms, were transferred intact from the unmanned simulation models when possible, and minimally modified when necessary to accommodate real-time processing constraints. Major system models and data bases were also transferred to the manned simulator as shown in Figure 2. Not only did this provide an environment in which behavioral performance of the aircraft systems and weapons was essentially the same between the two simulations, but it also had the benefit of allowing direct transfer of certain key data bases without the cumbersome and often fault-ridden process of reformatting. This ensured not only correlation of behavior but also correlation of system performance.

Many of the early problems in correlating the absolute magnitude of effectiveness results between simulations were caused by differences in ROEs and mission objectives. The problem of ROE and mission objective coordination between simulations was actually one of pilot adherence to the specified ROEs in the manned simulations. The pilots tended to ignore an ROE in order to improve their capability if the opportunity presented itself. This can be categorized as "gaming." The pilots, through multiple repetitions of simulated engagements, would become too knowledgeable of the opposing force, so they could (and would) disobey the ROEs to improve their competitive position.

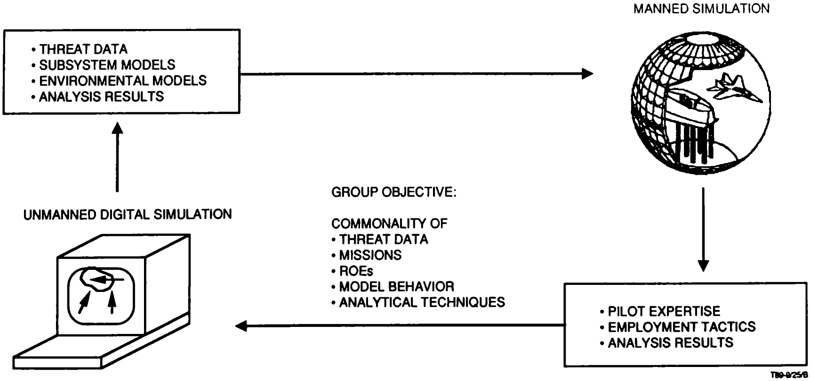


Fig. 2. CSEA Interface Group

Establishing ROEs and objectives for the pilots in the manned simulations and ensuring that they were followed became a critical task for achieving traceability of results between simulations. ROEs must be strict enough to minimize the gaming opportunities, but not so strict that the pilots are unable to explore the operational employment options that they are there to provide. This issue must be resolved each time a new simulation is conducted and often requires modifying ROEs prior to the actual start of the manned simulation investigation.

Simulation Enhancements

Through manned simulation comparisons and lessons learned, the fidelity of the digital pilot in unmanned simulations has been improved dramatically. For example, Tac Brawler production rules have become highly representative of actual pilot behavior. The tendency is to add more detailed actions and develop the capability to control more aspects of the pilot behavior and systems interfaces through production rules. However, the production rules can never simulate all of the dynamics observed in the manned simulation, nor is there any desire to do this. Manned simulation will always involve tactical experimentation on the pilot's part -- sometimes successful, sometimes not. The Tac Brawler production rules represent the results of the learning curve associated with this experimentation. Therefore, manned simulation results will capture, on occasion, the tails of the real-world distributions while Tac Brawler results will always fall closer to the mean.

In 1986 Northrop improved the manned simulation facility further by constructing an entirely new facility to house the existing manned simulation equipment plus an additional dome and seven entirely new MICSs. This facility was called the Integrated Systems and Simulation Laboratory (ISSL) (Figure 3). This gave Northrop the capability of including nine participants (the 24-ft fixed-base dome, the 28-ft fixed-base dome and seven MICSs) in the simulated air combat environment.

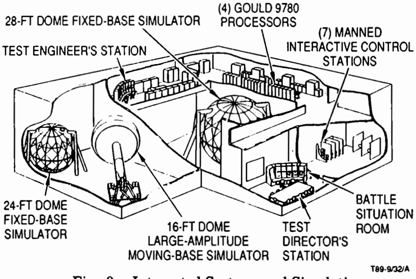

28-FT DOME FIXED-BASE SIMULATOR
(4) GOULD 9780 PROCESSORS
(7) MANNED INTERACTIVE CONTROL STATIONS
TEST ENGINEER'S STATION
24-FT DOME FIXED-BASE SIMULATOR
16-FT DOME LARGE-AMPLITUDE MOVING-BASE SIMULATOR
TEST DIRECTOR'S STATION
BATTLE SITUATION ROOM
T89-9/32/A

Fig. 3. Integrated Systems and Simulation Laboratory

Northrop continued its philosophy of designing for as many manned participants as possible and minimizing the number of digitally controlled participants. The new facility also provided new and more powerful processors to support the additional participants. These processors allow manned simulation designers to more easily introduce many of the environmental and subsystem models that are present in the unmanned simulation models. This facility, and the analytical lessons learned from the application of the cooperative simulation process over the preceding four years, provides the means for calibration and coordination of manned and unmanned simulation effectiveness analysis.

CSEA Application Examples

Example 1

An example of a concept development study is illustrated in Figure 4. The aircraft designers were required to quantify the potential contributions of certain key technologies or subsystems (hereafter referred to simply as technologies) toward mission effectiveness of an aircraft design concept. These technologies are typical of the type and problem space explored early in aircraft design work. The goal of this process is to ascertain whether the added effectiveness afforded by each of these technologies will justify the associated cost and weight penalties.

OPTIMIZATION FUNCTION: COST AND WEIGHT VS. EFFECTIVENESS

KEY TECHNOLOGIES: OBSERVABILITY LEVEL
WEAPONS CARRIAGE CAPACITY
VEHICLE PERFORMANCE
SENSOR TYPES
SENSOR INTEGRATION
STOVL

APPROACH:

1.) INVESTIGATE MAJORITY OF PROBLEM SPACE WITH UNMANNED SIMULATION
2.) INVESTIGATE SELECTED ALTERNATIVE CONCEPTS WITH MANNED SIMULATION

EXPECTED RESULTS:

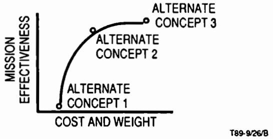


Fig. 4. Concept Development Example

Manned and unmanned simulations are applied to the extent that the full problem space (all possible combinations of technologies) is explored to the extent necessary by unmanned simulations while selected alternative aircraft concepts, using limited combinations of technologies, are evaluated using manned simulation. In this way, the advantage of unmanned simulation's capabilities to analyze larger problem spaces and provide better statistical validity is exploited. The manned simulation is utilized to explore the operational employment benefits of each technology and to provide substantiating data on pilot behavior to the unmanned simulation models.

For the example considered, the technologies were grouped to form the selected alternate concepts, as shown in Figure 5. In comparing the aggregate results between the alternate concepts, differences in observed mission effectiveness cannot be attributed to a single technology. It is possible to gain insight about the contributions of each by examining various low level figures of merit such as differences in system track file frequency and durations as measures of offensive sensor capability. Although this gives an indication of the relative contributions made to mission effectiveness, it can be inconclusive because of possible interdependencies between technologies.

Unmanned simulation analysis fills the gap between alternate concepts of the manned simulation data base by exploring the individual contributions of each technology. By structuring individual technology trades between alternative concepts, the individual contributions of each technology can be evaluated.

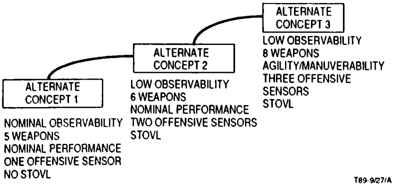


Fig. 5. Concept Technology Application

Figure 6 depicts the supporting analysis and concept refinement preceding the application of CSEA. After the prerequisite operations analysis composed of threat, mission and technology definition, preliminary unmanned simulations are run to initially explore the implications of each technology by varying each capability singularly.

The aircraft concept refinement process produced the alternative concepts of interest based on preliminary system requirements. These concepts are evaluated using manned and unmanned simulation with the CSEA approach. Manned simulations would be run to establish a basis for operational behavior of the alternate concepts defined by the technology assessments and initial unmanned simulations. Manned simulation results would be analyzed to understand the tactical implications of the various technologies and the resultant pilot behavior. Simultaneously, unmanned simulation of the alternate concepts would be run to provide a basis of comparison between simulations from which behavioral rules may be fashioned that better emulate the observed behavior in the manned simulations. It is often necessary to run the unmanned simulation behavioral rule development through several iterations before an acceptable level of behavior calibration is possible between the manned and unmanned simulations.

Exact duplication of all measures of mission effectiveness is never realistically possible due to statistical variations and the random behavioral deviations present in the manned simulation. If top level measures agree, such as kill and loss rates, and subsequent pilot review of unmanned simulation model behavior proves to be acceptable, the manned and unmanned simulation elements are considered to be synchronized within the problem space examined.

The unmanned model can then be applied freely to fill out the problem space matrix with confidence that it is accurately reflecting pilot operations of the technology in question. It can be used to establish the large statistical data base required in the analysis on which recommended design decisions will be based. Figure 7 shows sample results of this example if threat aircraft killed and friendly aircraft lost were the primary measures of mission effectiveness.

Example 2

As stated earlier, the unmanned simulation can and has been used to investigate the "whys" behind manned simulation results. An example is a case in which the manned results for an escort mission scenario showed low survivability for the escorted strike aircraft. It was felt that the low success rate was primarily a function of simulator limitations, which forced the strike formation to be represented as a welded diamond formation and also limited the number of escorts, reducing their employment flexibility.

In order to investigate these possibilities the same mission scenario was simulated using the AIRBAT unmanned model. AIRBAT was set up using data and assumptions from the manned simulation work. Three cases were explored. Case 1 was the baseline or "calibration case" which was identical to the manned simulation work, using strikers in a welded diamond formation and the same escort positioning. In Case 2, the strike package was simulated as individual aircraft in trail formation to give increased mutual support and defensive capability. This also created a more difficult target for the Red interceptors. Case 3 returned to the welded diamond formation, but added an additional Blue fighter flying close escort.

The results, shown in Figure 8, indicate that the Case 1 results track well with the manned simulation results. When the strikers were put in trail (Case 2) they tended to get more aircraft to the target, primarily because they can provide better mutual support. In Case 3, the addition of the close escort had the most dramatic effect, indicating a strong need for increased participants in the manned simulation. These results were used as the input for defining necessary improvements in the manned simulation capability and for improving the employment concepts used in the escort scenario.

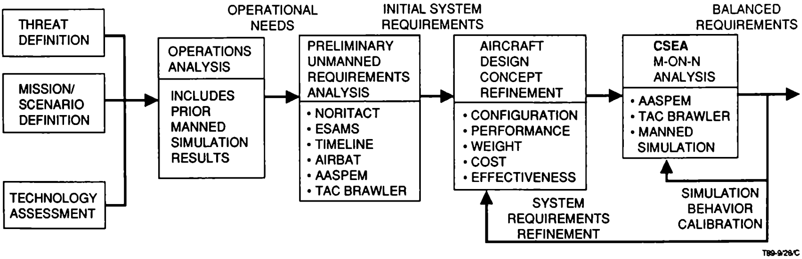


Fig. 6. Requirements Definition and Systems Analysis Process

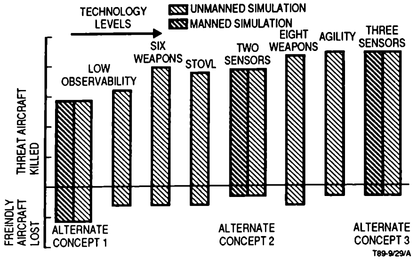


Fig. 7. Concept Development Mission Effectiveness Results

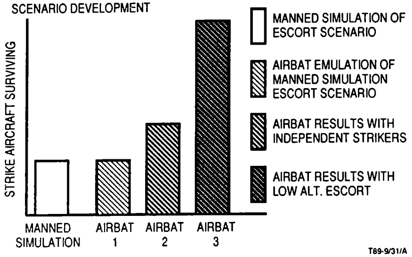


Fig. 8. Mission Scenario Development

Lessons Learned and Summary

The cooperative simulation approach has resulted in a highly synergistic approach for evaluating advanced aircraft concepts and technologies. Each type of simulation has its principal strengths which define the role each plays in the cooperative simulation approach.

Unmanned simulation, by way of its flexibility, repeatability, and ease of execution, is used to look at a broad range of issues. These unmanned simulation analyses act as a screening process from which the key issues, or driving technologies for mission effectiveness, can be identified and then in turn evaluated in the manned simulation. Additionally, technologies which are highly dependent on pilot decisions and interaction with the aircraft can be identified and earmarked for evaluation in the manned simulation.

The unmanned simulation provides the means for expanding on the manned simulation results. Since several factors may be changed simultaneously for the manned simulation test articles, the distinct cause of observed changes in effectiveness may not be clear. As shown, the unmanned simulation can be used to analyze intermediate points between manned simulation test articles. Additionally, the unmanned simulation models are easily accessible to the analyst to make changes and then rerun, permitting more experimentation in determining sensitivities to technology excursions.

The manned simulation cannot, by its nature, examine an extremely large problem space. The task of doing this will always rest with the unmanned simulation models. The contributions of manned simulations to CSEA process can best be understood by examining how information is gathered by an aircraft, transmitted to the pilot, and acted upon. This process can be depicted as a closed-loop transfer function through which all information regarding the aircraft's environment is passed. Figure 9 illustrates this transfer function for each of the major levels where information is acted on in a tactical environment.

The parameters that control the efficiency of the information transfer are shown in the middle of the transfer loop and the values which can be used to measure that efficiency at each step are shown as a product of each information level. The efficiency with which the information is passed is controlled by the technologies being investigated and the pilot's control of those technologies.

Manned simulation provides unique insights to the aircraft design and requirements analysis process through the measurement of situation awareness (SA) and development of new tactics.

The difficulty in measuring the contributions of the pilot's understanding of the tactical situation and his chosen tactics, and subsequently being able to distinguish between avionics technology contributions and pilot contributions, has been limited by the inability to actually measure SA. Unless SA can be measured, the only logical assumption is that some fixed percentage of all the information available in the aircraft is being acted on by the pilot. This has classically been a limitation of unmanned simulation since the SA of digital pilots is artificial. Using manned simulation, it becomes possible to observe a pilot's information processing and behavioral responses.

Measurement of his information processing capability or SA has recently become possible in manned simulations through application of the Situation Awareness Global Assessment (SAGAT) tool. SAGAT is a Northrop proprietary technique for quantitatively measuring SA and is described in detail in Reference Nos. 1–3. With SAGAT it is possible to measure a pilot's cognizance of key elements in his environment and rationalize his behavior in response to those elements. This rationalization then provides the basis of coordination between manned and unmanned simulations. Without this element, it would be very difficult to achieve correlation of pilot behavior between the two simulations.

By replicating as closely as possible the pilot's information processing, SA and tactical behavior in the unmanned simulation models, we can establish, from the cooperative simulation process, a combined data base which can be used to determine tactical weapon system requirements.

Neither manned nor unmanned simulations provide a complete tool for the evaluation of m-v-n engagements. However, together they do provide a comprehensive approach. The unmanned simulation provides the tool for system evaluations but, more importantly, is best at performing parametrics and partial differential technology evaluation for exploring sensitivities and trends. The manned simulation provides the real-world dynamics and pilot-vehicle interface and provides the means for introducing operational reality into the unmanned simulation. The combined results are far superior to what either can produce individually.

Northrop's success in applying the ISSL to the aircraft design process has driven the desire for further ISSL development. Northrop is currently expanding the ISSL through the addition of more participant stations and processing capacity.

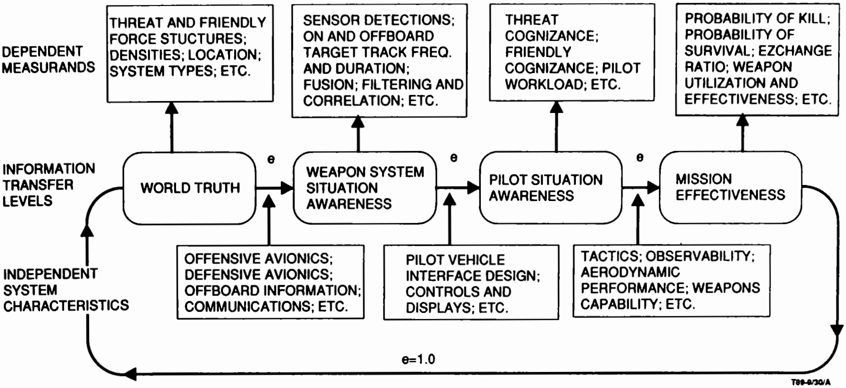

DEPENDENT MEASURANDS
THREAT AND FRIENDLY FORCE STUCTURES; DENSITIES; LOCATION; SYSTEM TYPES; ETC.
SENSOR DETECTIONS; ON AND OFFBOARD TARGET TRACK FREQ. AND DURATION; FUSION; FILTERING AND CORRELATION; ETC.
THREAT COGNIZANCE; FRIENDLY COGNIZANCE; PILOT WORKLOAD; ETC.
PROBABILITY OF KILL; PROBABILITY OF SURVIVAL; EZCHANGE RATIO; WEAPON UTILIZATION AND EFFECTIVENESS; ETC.
INFORMATION TRANSFER LEVELS
WORLD TRUTH
e
WEAPON SYSTEM SITUATION AWARENESS
e
PILOT SITUATION AWARENESS
e
MISSION EFFECTIVENESS
INDEPENDENT SYSTEM CHARACTERISTICS
OFFENSIVE AVIONICS; DEFENSIVE AVIONICS; OFFBOARD INFORMATION; COMMUNICATIONS; ETC.
PILOT VEHICLE INTERFACE DESIGN; CONTROLS AND DISPLAYS; ETC.
TACTICS; OBSERVABILITY; AERODYNAMIC PERFORMANCE; WEAPONS CAPABILITY; ETC.
e=1.0
T89-9/30/A

Fig. 9. Information Transfer Function

The new capability will allow for 13 manned participants and the capability to support extremely high fidelity environmental and subsystem models for all players. This will greatly enhance Northrop's ability to support new advanced aircraft and technology development studies in the future.

Acknowledgement

The authors would like to thank Mr. Jack Gibson, Mr. Eiichi Kamiya, Mrs. Barbara Householder, and Mrs. Susan Blancett for their assistance in the editing of this paper. Their contributions improved the continuity, integrity, and substance of the text and have been greatly appreciated.

References

1. Hutcheson, J.H., and Segerblom, R.L., "TACTICS: A Three Body, Three Dimensional Intercept Simulation Program," Rand Corporation, October 1969.

2. Luhn, J., "Tactical Air Combat in a Real-Time Multiple Engagement Simulation," AIAA Flight Simulation Technologies Conference, No. 88-4601, Atlanta, Georgia, 1988.

3. Endsley, M., "SAGAT: A Methodology for the Measurement of Situation Awareness," Northrop Technical Report NOR 87-83, August 1987a.

4. Endsley, M., "Preliminary SAGAT Validation Study," Northrop Technical Report NOR 87-120, December 1987b.

5. Endsley, M., "Situation Awareness Global Assessment Technique (SAGAT)," Northrop Technical Report NOR 88-26, May 1988. Presented at the National Aerospace and Electronics Conference (NAECON), Dayton, Ohio.

FULL FIELD OF VIEW DOME DISPLAY SYSTEM

Captain Brian A. Reno

Air Force Human Resources Laboratory
Williams Air Force Base, Arizona

Abstract

The requirements for a visual system to adequately support a fighter aircraft's missions, particularly the low level and air-to-ground scenarios, are very demanding. Present day visual systems lack the brightness, field of view, and/or resolution to satisfy these requirements. The Full Field of View Dome Display System, being developed by McDonnell Douglas under contract from AFHRL and ASD/YW, is an effort to address this problem by providing a display system with higher brightness and resolution than previously attained in a dome simulator over 100% of the field of view of a modern day fighter.

Introduction

There is very little opportunity for the modern day fighter pilot to fly as he truly would in a wartime environment. Safety restrictions, airspace limitations, and civilian complaints/concerns are among the many factors limiting today's training. Recent mishaps, especially in the European theater, have spawned additional restraints in the amount and type of tactical air combat training. There is growing concern that with these constraints, today's pilots are not receiving adequate and sufficient training in the type of flying he is expected to do during war. A simulator that can effectively supplement or enhance the flying time of the fighter pilot is a much needed commodity. An excellent example of this is the German Air Force's commitment of approximately $400 million dollars for the procurement of a number of simulators for the Tornado aircraft, aimed specifically at alleviating this situation.

One of the primary approaches the Germans are examining involves a fiber optic helmet mounted display system. While these types of devices possess certain advantages over dome displays, (namely facility size requirements, brightness, and multiple crew aircraft simulation potential), they are not necessarily the best display systems for a fighter aircraft. Problems exist in detail image quality due to the fiber optic bundle structure. The additional optics that must be worn are a disadvantage both physiologically and operationally. Dome systems, however, do not require such extensive hardware on the helmet. A much greater stabilized instantaneous field of view is possible with a dome display system. Finally, domes currently are the more accepted display system in the simulator community. The Full Field of View Dome will be an excellent tool for AFHRL/OT to explore these tradeoffs.

Background/History

The Initial Approach:Variable Acuity/Eye Tracking

Variable Acuity Optics

Variable acuity (VA) optics are an eloquent engineering solution to two major problems facing domed simulators today, namely low resolution and high imagery cost. Quite simply, VA optics allow the image generator to match the resolution capability of the human eye.

The human eye has an average acuity of approximately 1 arc min on axis which rapidly drops off as a function of distance from point of fixation as seen in figure 1. If the computer image generator geometrically distorts the image, and the image is projected using specially designed non-linear (VA) optics, the pixels on a linear light valve format are redistributed or "packed", so as to match the eye resolution curve, resulting in a variable resolution display (1).

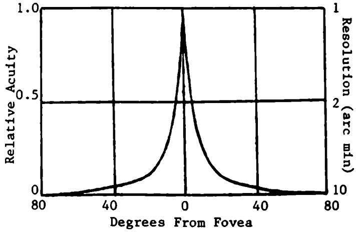


Fig 1 - Human Eye Acuity

There are two significant advantages VA optics possess over conventional linear optics. Apparent resolution and scene uniformity are greatly improved due to a smooth resolution falloff. This is in contrast to current high resolution inset domes which have a relatively sharp boundary area which is easily discerned by the pilot. Additionally, an enormous potential cost savings exists due to conservation of pixels which reduces the demand on the computer image generator's size and capabilities. The human eye can only resolve approximately 130,000 resolution elements or pixels at any given time (1), whereas a light valve or similar system can project a significantly greater amount. If used efficiently, as is done using VA optics, the number of Image Generator (IG) channels required for total field of view and high resolution is dramatically reduced.

Oculometer/Eye Tracking

VA optics technology when applied to simulation relies heavily upon the ability to accurately and quickly align the projection with the pilot's point of gaze. The rapid resolution falloff of VA optics demands that the projection and the pilot's point of gaze be aligned to less than one degree under the dynamic conditions of tactical combat. Head movement, coupled with eye saccades and pursuits combine to make this a formidable technical challenge which is yet unresolved. Current oculometer or eye tracking technology cannot track the eye to such a degree of precision and required stability (noise induces variances in eye position data which results in image "jitter" - an unacceptable situation). Additionally, the human eye is capable of accelerations of approximately 40,000 deg/sec/sec and linear velocities of up to 700 deg/sec (2). MCAIR has designed a servo system capable of up to 53,000 deg/sec/sec accelerations and linear velocities approaching 1,000 deg/sec. Taking advantage of saccadic suppression (a phenomenon in which the brain suppresses the visual scene during rapid eye movements or saccades) and prediction algorithms, the servo system can "close" to the eye's position in the necessary time. This servo system is incorporated into the current linear optics design which will allow a relatively easy retrofit to VA optics and/or eye tracking when this technology has matured sufficiently.

The Current Approach: Linear Optics

Due to the shortcomings in current oculometer technology, in December 1987 the design was modified deleting variable acuity and eye tracking. The current approach incorporates conventional linear optics, utilizing a total of seven IG channels - six background and one area of interest (AOI), which is either target or head tracked.

Current Design

Overview

The Full Field of View Dome Display System incorporates a 24 foot high gain dome surface, six channels of background imagery, and one target or head tracked linear inset (see figure 2). 100% of the field of view of an F-16C will be displayed. The projector lenses are located immediately above and behind the pilot's head, and are out of view during normal head and body movements. Imagery is projected onto the surface using General Electric 1000 lumen light valves. Computer imagery is generated using the Advanced Visual Technology System (AVTS), which is similar to a Compu-Scene IV image generator. An innovative video blending scheme, primarily accomplished through software, is incorporated to handle the demanding task of both static and dynamic blending of seven channels of projected imagery. The cockpit sits on rails and can be "rolled" out of the dome through a door in the lower front quadrant (out of the field of view of the pilot), allowing relatively easy maintenance or replacement of the cockpit. Railings on the platform around the cockpit will fold out of view when the simulator is in operation.

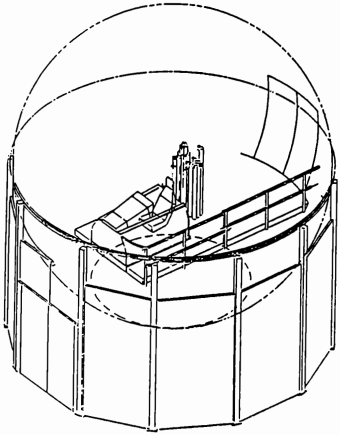


Fig 2 - Full Field of View
Dome Display System

Dome

A number of different dome sizes were considered. A 24 foot dome is being used due to size and brightness tradeoffs. In order to maximize brightness (10 ft lamberts design goal), a high gain screen is needed. There are, however, a number of tradeoffs associated with using a high gain screen. Although the apparent brightness is significantly greater, large variances can occur if the viewing point and projectors are not conjugate about the dome center. This fact imposes some critical design considerations on projector and cockpit location as well as the reflective properties of the dome surface. Considerable mathematical analysis was performed to optimize these variables. With a surface gain of approximately 10, a reflectivity of 0.58, and a reflected lobe shape modeled by the expression $(\cos \beta)^{38}$, projector and pilot location could be located so as to have acceptable brightness levels and uniformity, with negligible brightness falloff for head movement within the cockpit envelope. With this determined, the remaining obstacle was to find a dome surface with these properties. A number of paints were evaluated, and a commercially available aluminum paint was found that closely matched these characteristics.

Construction of the dome requires considerable precision and care due to the characteristics of the reflective surface. Joints, rivets, and irregularities in the surface or paint will be highly noticeable, and must be virtually eliminated if acceptable performance is to be obtained. Using a procedure not unlike painting a car, but with more steps and greater precision to insure a smooth and uniform surface, acceptable results are obtained.

With a full field of view, even seemingly minor design details must be considered. The entry and exit is located at the rear of the dome, and consists of a single door that is mated as closely as possible to minimize its visibility when the system is running. The platform and railings around the cockpit must be out of the field of view and still conform to safety codes. This was accomplished by positioning the platform well below the cockpit (the cockpit is situated so the pilot's head is very close to the dome center - approximately 12 feet in the air) until the minimum width as per safety codes was outside the cockpit field of view. The railings will fold inward and/or downward so as to also be out of the field of view during sim operation. Provisions are included so the railings automatically come up should the pilot have to egress the simulator.

Projectors

Each of the seven projector assemblies consists of a General Electric 1000 lumen light valve located underneath the cockpit platform, an optical chain or "light pipe" carrying the imagery into the dome, and a projection lens assembly The projection lenses are located above and behind the pilot so as to provide the greatest possible brightness capability (figures 2 and 3). The location of the Area of Interest (AOI) projector is the most critical in nature, and is positioned conjugate to the viewpoint about the dome center. The other projectors are oriented around the AOI projector providing the most advantageous tradeoff in coverage and brightness. The total width of the projector columns is less than 24 inches, and will not block the pilot's rear view significantly due to his body motion when "checking six".

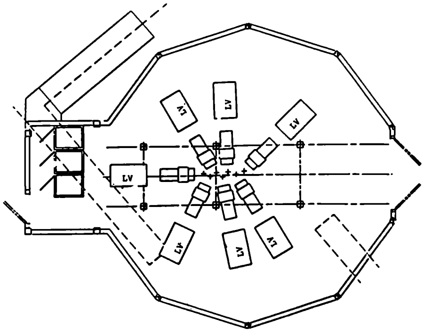


Fig 3 - Projector Locations

Background Projector Assembly

There are six background projector assemblies - a forward projector, left and right side projectors, left and right rear projectors, and an overhead projector. Their locations were optimized for resolution and brightness based on anticipated viewing time by the pilot. In other words, the forward projector was deemed the most important, the left and right sides next, etc. The area of coverage for each projector is approximately 72 degrees horizontally by 100 degrees vertically. The overhead projector covers a roughly circular area of approximately 50 degrees in radius. The maximum apparent brightness of the system at various points is shown in figure 4. Resolution for the background averages about 7 arc min.

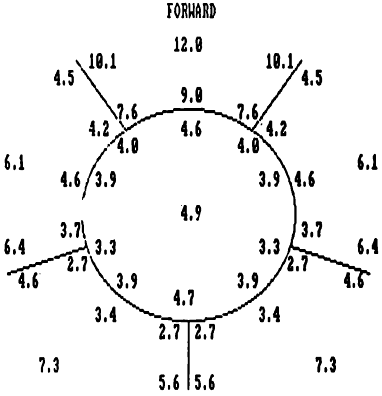


Fig 4 - Apparent Brightness in Foot-lamberts

The optics chains for the background projectors consist of a decollimating relay lens, several extension relay lens assemblies, and a fixed roll adjusting assembly. The total length of each optics chain approaches 17 feet, of which approximately 13 feet is in the vertical. The support structure for the optics was designed to minimize the width (pilot visibility considerations) while still providing adequate stability and ease of alignment.

Area of Interest (AOI) Projector Assembly

The optics for the AOI projector are similar to the background projectors with a couple of notable exceptions. The AOI is slewable over the entire dome surface, which necessitates dynamic roll adjustment. The servo systems to accomplish this must be capable of accelerations of 6000 deg/sec/sec and linear velocities of 360 deg/sec. MCAIR has designed the servos to handle accelerations of 53,000 deg/sec/sec and velocities of 1,000 deg/sec, which will allow the ability to forward fit eye tracking capability. The projector lens itself is a commercially available lens (as are the background projector lenses which is interchangeable for various projection field sizes. A 25 degree or 40 degree inset size are the options being delivered. With the 25 degree size, AOI resolution is approximately 2 arc min, and with the 40 degree inset resolution is 3.3 arc min.

Video Blending

The blending of six background channels and a moving AOI is a formidable challenge which MCAIR has approached in a very unique way. Predominately through software, both zonal (where two or more projectors overlap) and global (over the entire projected surface) adjustments can be made. This software blending scheme can successfully blend even the toughest "corners" of the dome - i.e. where three background projectors meet. This system also gives the dome great flexibility to the researcher by allowing him to vary the brightness of any area so as to achieve either maximum brightness, greatest uniformity, or any configuration in between these two extremes.

Background to Background Blending

The actual brightness profile of the dome without blending is represented in figure 5. A zonal correction is applied to smooth the areas of projector overlap. This is a purely static blending procedure, and results in a brightness profile illustrated in figure 6. From this profile a global correction takes place. Color adjustment, brightness profile shaping to include smoothing the peaks if desired, and compensation for screen gain variance based upon the current head position are all accomplished, resulting in figure 7's brightness profile. The global correction is a semi-static correction, with the only dynamic input being the head position affecting the screen gain. As can be seen, tremendous flexibility in the overall brightness profile is possible.

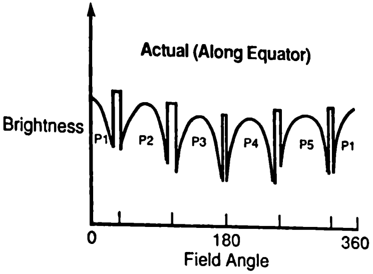


Fig 5 - Actual Brightness Profile

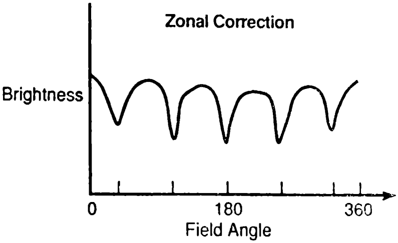


Fig 5 - Zonal Corrections

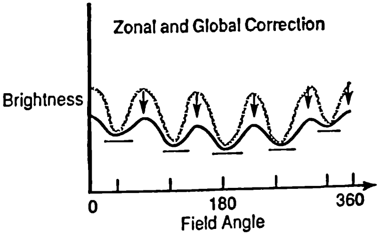


Fig 7 - Zonal and Global Corrections

Background to Inset Blending

In order to successfully blend the AOI with the background, a hole must be "cut" into the background imagery for the inset imagery. This involves blacking out the area for the inset and creating a 3 degree blend zone of varying intensity to facilitate a smooth transition between background and inset. This is a dynamic correction, as the position of the blend zone is constantly changing.

Inset to Background Blending

Both zonal and global corrections are also performed on the AOI. The zonal correction involves a static cropping of the imagery to "fit" into the hole created by the background to inset blending algorithm, and the creation of a blend zone which when projected over the background blend zone results in a uniform brightness. The global (semi-static) correction performed on the AOI is analogous to the background global correction with the exception of greater brightness reduction to match the background intensity. The final result is a blended AOI which follows the brightness profile of the dome as described by the user.

Conclusion

The Full Field of View Dome Display System will be located at the Air Force Human Resources Laboratory, Operations Training Division at Williams Air Force Base, Arizona. Installation of the dome will begin in September 1989, with completion scheduled for May 1990. It will be configured as an F-16, with the ability to change the cockpit to an F-15 configuration.

The Full Field of View Dome will possess greater capabilities than previous dome systems, providing the pilot with a more realistic environment, and the researcher with greater flexibility and more tools to work with. These capabilities will allow a number of research questions to be addressed. Brightness, color, scene content, and field of view requirements can be examined to a degree not previously possible. Comparisons of various types of visual systems can be made. The effectiveness and requirements of a system for aerial combat training, both in the air-to-air and air-to-ground, low level roles, can better be determined. The Full Field of View Dome will indeed be a valuable research device.

References

1. Fisher, Ralph W., "Design of an Optimal Simulator Visual System". Proc. of 8th Interservice / Industry Training Systems Conference, 18-20 Nov 1986. Salt Lake City, Utah.

2. Bahill, A., and Stark, L., "Trajectories of Saccadic Eye Movements". Scientific American, 240, 108-117, 1979.

89-3317-CP

A UNIVERSAL PROJECTOR FOR SIMULATORS DISPLAYS

Ralph W. Fisher
Evans & Sutherland
Salt Lake City, Utah

ABSTRACT

The search for a higher performance - lower cost simulator visual system has resulted in proliferation of numerous "Area of Interest" projection display concepts. Of the many Area of Interest concepts that now exist, there is no clear choice which will emerge as the best system for the future. To minimize the procurement risk associated with advanced visual systems the Simulation Division of Evans and Sutherland developed a "Universal Projector". This projector can be utilized in any Area of Interest configuration ranging from a simple target projector to an eye controlled Variable Acuity visual system. This paper discusses the design of this projector and describes how it is applied to each of the target, head controlled, and eye controlled concepts.

The resulting design shows that the Universal Projector has no size or performance penalties and can be converted from one concept to another by interchanging one or two optical modules and reprogramming control software. The flexibility and wide range of applications should make this projector very attractive to customers who do not want to become committed to unproven and costly display concepts.

INTRODUCTION

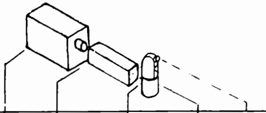
Simulator display technology is driven by the desire to fully support human vision at reasonable cost. The conventional approach of covering the entire visual field with eye limited detail is now recognized as an impossible task from both technical and cost standpoints. As a result the industry has seen the proliferation of numerous display concepts that attempt to take advantage of the variable acuity nature of human vision, i.e., high resolution is needed only in areas where the observer is looking. All advanced concepts for full field of view employ one or more slewable projectors that provide high acuity in the Area of Interest (AOI). These generally fall into two categories - those that utilize a fixed low resolution background and those that move the background with the AOI. In the first category we have target projectors, head pointed AOI, eye pointed AOI, and the eye pointed variable acuity inset. In the second category are head pointed dual field systems, and full field variable acuity. All of these utilize the same optical source, one or more light valve TV projectors. They differ only in the fields of view they project and speed of response. This suggests that any of these concepts could be mechanized with the same gimballed platform having interchangeable projection lenses. This would be very attractive because all the advanced AOI concepts beyond the target projector have various psycho-physical and technical problems that preclude a clear choice at this time.

The development of the Universal Projector was organized to provide minimum technical risk and cost. This was accomplished in two ways. First, a phased approach was used that established feasibility before commitment was made for detailed design and fabrication. Also, an initial application was selected that has minimal technical requirements and an existing market - the target projector.

DESIGN PHILOSOPHY AND REQUIREMENTS

The Universal Projector consists of the common platform, application specific projection lenses, and source coupling optics as shown in Figure 1. Requirements are also summarized in this figure. Note that the highest angular resolution is required for the target projector, widest field of view (FOV) for the slewable background configuration or full field variable acuity, and fastest gimbal response for the eye controlled applications. An additional requirement for all applications is to place as much of the light valve source output on the viewing screen as possible (Maximum Display Brightness). This requirement plus the desire to make the projector as small as possible provides the starting point which defines logic for the universal projector design.

FIGURE 1 THE UNIVERSAL PROJECTOR



CONFIGURATION	SOURCE	COUPLING OPTICS	PROJECTOR (CONTROL)	PROJ. FOV	LENS RES.
TARGET	LIGHT VALVE	5/1 ZOOM	COMPUTER 9000°/sec^2	20°	.5' ARC
SLEWABLE INSET (ESPRIT)	LIGHT VALVE	RELAY	HEAD/EYE 44000°/SEC^2	18°	2.4' ARC
SLEWABLE INSET & BACKGROUND	2 LIGHT VALUES	SCALING AND COMBINING	HEAD/EYE 44000°/SEC^2	40° & 120°	3.5' ARC & 10' ARC
VARIABLE ACUITY INSET	LIGHT VALVE	20/1 DISTORTION	EYE 44000°/SEC^2	40°	.75' ARC
FULL FIELD VARIABLE ACUITY	LIGHT VALVE	50/1 DISTORTION	EYE 44000°/SEC^2	180°	1.5' ARC

Another factor in design philosophy that had to be established before any serious design effort could begin was the way of handling projector offset from the dome center. The FOV requirements shown on Figure 1 are measured from the viewing point - usually the dome center. In many applications the projector will be displaced as much as half way to the dome surface. The resulting variable projection FOV with pointing angles can be accomplished in either of two ways. A projection lens can be selected for the maximum FOV required and then the image size optically reduced for smaller FOVs. The other way is to utilize a zoom lens for the projection lens. The latter is a clear choice when the impact on projector module size is considered. The optical relay that conducts the light through the projector must be much larger than necessary and have higher optical quality (by a factor of 3) to support the fixed lens. The projector overall size would have to increase by at least this amount. On the other hand the zoom lens is not much larger than a fixed lens.

Another benefit that arises from the selected approach is that the relay resolution can degrade from center to edge - a characteristic of all optical relays that can only be overcome through large element sizes. For example, the very high resolution needed for the target projector must exist over only 1/5 of the relay image plane. The remainder of the image supports wider fields of view and therefore can have lesser resolution. A similar situation exists for Variable Acuity and the wide field slewable configurations.

PRELIMINARY DESIGN

Design can be logically broken down to the universal portion that is common to all applications and the application specific portion.

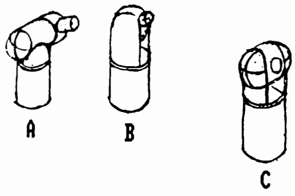
The Universal Portion

The universal portion of the Universal Projector is shown as the Projector in Figure 1. It is the most complex of the modules and was the most difficult design task. It contains the optical relay that couples the light through the gimbal axes to the projection lens. The size of the projector will be wholly determined by the size of this optical relay. Therefore, it is very important to keep this light path as small in diameter as possible.

The Optical Relay. A small optical relay is not consistent with good optical performance. Small size is achieved only through many relay stages which means lots of glass which relates to light loss, quality reduction, and high cost. Extensive tradeoffs were conducted to determine the optimum relay configuration. Results showed diameters in the 50 to 60 mm range gave the best balance between size, efficiency, and optical quality. In addition this size range was most compatible with both the light valve source and off-the-shelf 35 mm camera lenses.

Using these constraints on relay geometry, a preliminary layout of the projector was possible. Three configurations were conceived. These are shown in Figure 2. They differ in the gimbal arrangement relative to the optical relay.

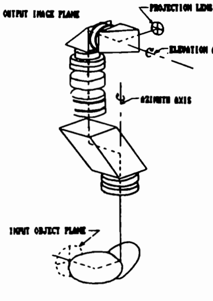
FIGURE 2 PROJECTOR CONFIGURATIONS



Configuration A is simplest optically but has considerable asymmetrical mass and optical offset from the gimbal axis. Configuration C has the best symmetry but is very complicated optically and mechanically. A compromise between these two was selected, Configuration B.

A first order optical design was conducted to establish image sizes, spacings, and Fnumbers. This was a highly integrated effort with the mechanical design because only certain areas within the projector module are available for optical components. First order analyses are very good at establishing this geometry but tell nothing about image quality. In order to study this the relay was set up in the laboratory using off the shelf optical components. Because exact matches for the relay lenses could not be located, optical equivalents were constructed. Relay performance was found to be adequate for all applications however - it was clear that a custom optical design would be required to fit into the envelope requirements. The final custom design is shown in Figure 3.

FIGURE 3 PROJECTOR RELAY OPTICS



Control System. Once the optical approach for the Universal Projector was defined, conceptual design of the hardware was possible. Moments of inertia were estimated and motors sized for gimbal drives. The worst case response (eye control) was used for this. We found direct drive brush type motors were required to meet our response and space requirement. Space and accuracy requirements dictated use of incremental encoders for elevation and azimuth position sensing. Servo analyses showed that both gimbal drives would require current command rather than voltage command to eliminate the back emf, damping, and winding inductance effects in order to achieve this response with the required stability and insensitivity to parameter variations. The analyses also revealed that the desired response could only be achieved with an intelligent control system that prevents over-driving or saturation of drive electronics. This same control system must also have the capability of shaping servo inputs as required by the less demanding applications. A microprocessor based control system was selected to provide the required capability.

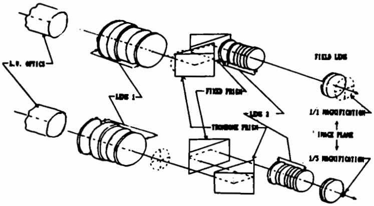
Application Specific Design

For the target projector application this consists of the projection lens and the coupling optics between the light valve and the gimballed projector.

The Projection Lens. The projection lens must maintain a 20 degree field of view at the dome center for all gimbal angels and projector locations. This requires a 3/1 zoom lens because of the projectors' off dome center location. We decided to use an off-the-shelf 35 mm camera lens here because of their high optical quality, low cost, and small size. Because this lens is manually operated in both focus and zoom, a servo drive had to be designed. Here we elected to drive the zoom cam of the lens and add an additional cam for focus (which in our case is always a function of zoom). The extremely tight packaging requirement and high internal friction of the zoom mechanization dictated a gear drive mechanism with 9/1 ratio. The requirement to keep up with maximum gimbal travel and hold against friction without overheating sized the motor. Servo analyses showed that the encoder had to be directly coupled to the motor shaft to achieve the required accuracy and dynamic response. A unique position reference is achieved with this arrangement because focus cam travel is about 60 degrees which holds motor rotation well less than 2 revolutions.

Coupling Optics. For the target projector this consists of 5/1 zoom optics. Here we elected to utilize a variable conjugate type of zoom optics i.e., a single lens with variable object and image distances. The main reason for this was it's simplicity and ease in separating and controlling both image and pupil positions separately, a requirement unique to the optical construction of the G.E. light valve. The elements of this zoomer are shown in Figure 4. The actual zoom is achieved by motions of lens 2 and the trombone prism. The pupil control is achieved through lens 1 plus an E & S proprietary "Pupil Scrambler." The pupil scrambler operates in conjunction with lens 1 to provide a color balanced light valve output coupling into the downstream optics. This approach is very near 100% efficient in maintaining constant display brightness and color. This can be compared to the inefficient devices that are normally used for this function such as fiber optic plates or diffusers which result in low light output and/or poor optical quality.

FIGURE 4 5/1 ZOOM COUPLING OPTICS



The 5/1 zoom configuration was set up in the laboratory and evaluated for color balance and optical quality. In this case suitable off the shelf relay lenses were found for lens 2 but a suitable light valve coupler (lens 1) could not be found and required an "optical equivalent" as in the projector module case. Performance of these optics were found to be excellent.

The required mechanical element motions as a function of zoom were achieved by a single drive cam. An extensive analysis was required to define the cam shapes and drive requirements. In this case the maximum closing rate and minimum separation of the simulated and target aircraft established peak dynamic requirements.

Experimental Verification

Items identified in the first order analysis that needed experimental verification were optical quality and servo performance. Questions on optical quality were resolution of the total system and effectiveness of the pupil control technique in maintaining color balance in the projected display. For the servo control system, the basic concept needed to be validated.

Optical Experiments. The entire optical system was set up in the laboratory using the first order designs discussed above. A G.E. light valve was used as the source and several 35 mm camera zoom lenses were evaluated for use as the projection lens. A Nikon lens was selected as the one most suitable for our application. This lens had far better performance than required, was inexpensive and could be most easily adapted to our drive requirements.

Both resolution and color balance were assessed over the entire 15/1 zoom range possible. Results were excellent except in the widest field condition where resolution deteriorated at the field edge. The cause was classic field curvature inherent in relay optics. It was clear that a custom lens design would be required to correct this problem. The pupil control technique worked perfectly. It maintained resolution, brightness, and contrast in the projected display over the entire zoom range.

Servo Control Experiments. The elevation servo motor was set up in the laboratory with a dummy mass to give the correct load inertia. The control system was breadboarded using a 68020 microprocessor. Everything except the lead compensator and, of course, the drive amplifier was mechanized within the microprocessor. Servo response was exactly as predicted in the simulations.

DETAILED DESIGN

The Detailed Design is logically broken down into Optical, Mechanical, and Electrical design areas.

Optical Design

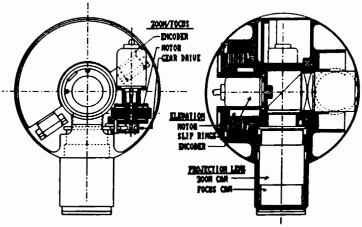
The first order optical designs discussed above served as input for a detailed optical design. After 3 iterations the designs shown in Figures 3 and 4 evolved. Most of the effort was expended on the "Universal" portion, Figure 3, because these optics must perform well in all applications. Modulation is 50% or better at the raster limiting spatial frequency in all cases. Minor vignetting occurs only in the very large field of the slewable background configuration. Since no scene matching at the edge of the wide field of view is required with this configuration the effects of vignetting will be non noticeable.

When the entire target projector is assessed including the light valve and projection lens, modulation at the raster spatial frequency limit is better than 50% over the entire raster at maximum zoom (smallest projected field) and 40% minimum zoom.

Mechanical Design

The mechanical design of the most complex portion of the Universal Projector will be explained with help of Figure 5 and 6. Figure 5 shows the gimballed platform and 3/1 zoom

FIGURE 5 ELEVATION ASSEMBLY



lens mechanization within the projection head. The 55 oz-in motor used on this zoom drive, the focus cam, gear drive, and encoder are notated on this figure. A Cannon encoder was used here because its superior performance and packaging features. This encoder is common to the elevation, focus, and zoom axis. The gimbal acceleration design goal of 44,000 degrees/sec ^2 was achieved with a 5 ft-lb drive motor on elevation and a 25 ft-lb motor on the azimuth axis. The Cannon encoder was installed for elevation position sensing as shown on Figure 5. A Teledyne Gurley modular incremental encoder was designed into the azimuth axis concentric with the optical path as shown on Figure 6.

FIGURE 6 AZIMUTH ASSEMBLY

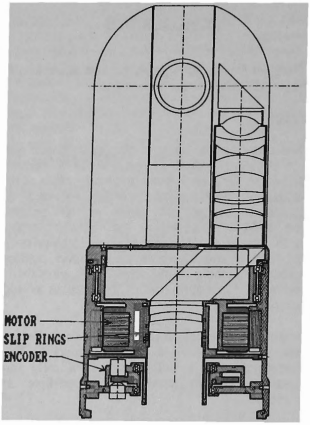
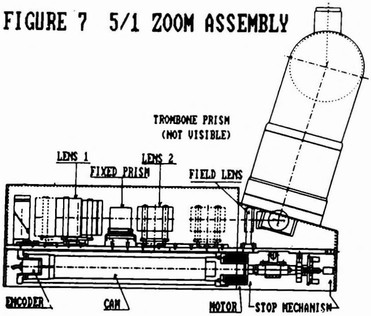


FIGURE 7 5/1 ZOOM ASSEMBLY



The remainder of the design is the light valve coupler which for the target projector consists of the 5/1 conjugate zoomer. This design is shown in Figure 7. Element motion requirements are shown versus zoom on the lower portion of the figure. All translating elements are mounted on ball slides. These are driven by cam followers that ride in grooves in the servo driven cylindrical cam. A 5 ft-lb motor was required to meet the worst case flyby conditions. Again the Cannon encoder was used for position sensing. One of the most difficult design tasks on this module was protecting against a runaway servo. The masses of the elements are quite heavy and move very rapidly during normal operation and very little over-travel is possible for engaging hard stops. A centrifugal velocity limit plus a screw driven stop mechanism with energy absorbing bumpers was required.

Figure 8 is a photograph of the complete projector and its light valve source. Covers are removed to show some mechanical detail in Figure 9.

FIGURE 8 PHOTO OF TARGET PROJECTOR

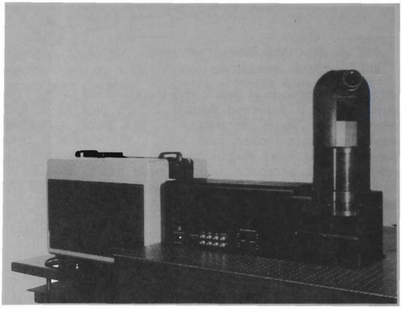
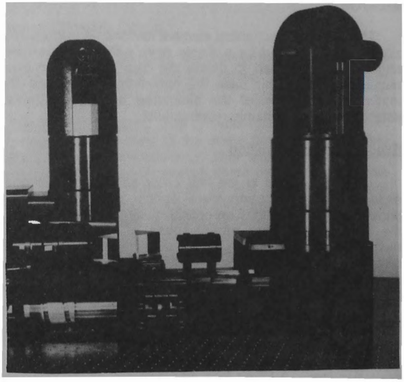


FIGURE 9 PHOTO WITH COVERS REMOVED



Electrical Design

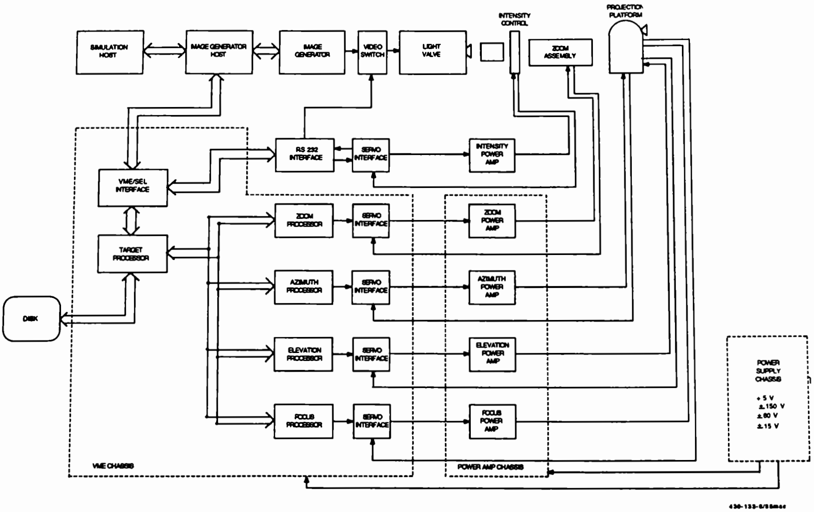
The detailed electrical design consists of the electronics and associated hardware necessary to control the projector and interface with the host processor. A simplified block diagram of the electrical system is presented in Figure 10. Because of the high response of the gimbal servos no mechanical stops could be considered on either gimbal axes. Slip rings are used to couple drive signals and control signals across these axes. They couple drive signals in and control signals out of the gimballed projector assembly. To provide maximum noise suppression the motor drive signals are physically separated from the control signals and twisted pair wiring is used on both. In addition, because of the critical nature of the incremental encoder operation, its pulses are counted and converted to digital position words as close to the encoder as possible. This position is then serially sent out through the slip rings over the control lines. The clock rate on this serial transmission is 2 Mhz .

In order to support handoff between target projectors, high speed video switching is provided that is synchronized with vertical retrace.

CONCLUSION

The goal of this development has been to provide a high performance color target projector competitive with existing target projectors. Included in the design of this projector is an inherent growth path that allows the projector to assume any of the target slaved, head slaved, or eye tracked systems with only minor hardware and software configuration changes.. This flexibility provides compatibility with present day display technology and inherent flexibility to upgrade as the technology for eye tracking and variable acuity matures. These upgrades can be accomplished from the common projection platform presented in this paper.

FIGURE 10 TARGET PROJECTOR FUNCTIONAL BLOCK DIAGRAM



89-3318-CP

INCORPORATION OF TARGET PROJECTORS FOR WITHIN VISUAL RANGE SIMULATION

Kenneth R. Henke*
Sr. Research & Development Engineer
Lockheed Aeronautical Systems Company
Burbank, California

Abstract

This paper discusses the technical requirements for servo driven target image projectors used in air-to-air tactical simulation. It covers the design and implementation of two different systems installed in the Weapon System Simulation Center of the Lockheed Aeronautical Systems Company.

Background

Simulation of an air-to-air combat scenario for the within visual range scenario places requirements on the visual simulation that are difficult to meet under the present state of the art capability. Air-to-air combat requires a visual field of view (FOV) that is extremely large. A reasonable range for this FOV is the complete upper hemisphere plus the areas of the lower hemisphere from approximately -25° over the aircraft nose, to -45° at the sides, and up to the horizon line over the aircraft tail. This large FOV requirement calls for either a head/eye slaved background visual system or a multi-projector system with a resultant relatively low resolution background scene. At Lockheed, the latter mechanization, using four background projectors, provides a background resolution ranging from 8 to 11 arc-minutes per TV line. While this resolution is adequate for general flight operations, it is not adequate for within visual range combat where resolution consistent with the pilot's visual acuity is required.

Such a capability has been provided in the past through the use of super-imposed high resolution target images on the dome screen surface. While this approach results in some severe limitations such as excessive brightness, bleed through of the background image, etc., Lockheed chose to use this method over an inset image approach as the least risky although the desired performance improvements on the existing mechanizations still required a new design and mechanization.

The following discussion covers the determination of requirements for such a system, the design of the system, and the integration problems observed in implementing two different versions of target projector systems at Lockheed.

General Requirements

The dome size, cockpit installation, physical projector size and dome screen gain characteristics all affect the target projector installation as well as the number of target images to be presented in the simulation scenario.

For several reasons, Lockheed has chosen to provide their simulation in a 28 ft. diameter dome with a gain screen (gain of 2). This affects the real estate available for installation of target projectors and impacts the apparent brightness of the target images due to the target projector/ screen/pilot angles.

The use of the General Electric Talaria projector as the image source for the target projectors provides a sufficiently high resolution and bright image that it allows the use of a half-raster split image for each individual target image. This cuts down on the number of individual CIG channels needed for the target images and the number of light valves or CRTs to be located within the dome.

Optimally, all target image projectors should be installed at the center of the simulation dome. Unfortunately, this is precisely where the pilot sits in the simulation cockpit. The laws of physics have not been repealed and it is necessary to compromise the location of the target projector heads. It was determined that locations for four panoramic heads could be found at the four corners of the cockpit installation. The heads were located just under the cockpit visible FOV line.

The same basic configuration was utilized in the second system to be installed in the Lockheed Weapon System Simulation Center (WSSC). In this case another set of 4 panoramic heads were installed in positions directly outboard from the initial 4 heads. This provided capability for two more target images throughout the dome while not causing any visible shadowing of the existing head installations. The additional heads do suffer from the effects of greater head/screen angles on distortion and screen gain effects. These effects are considered to be within an acceptable range, however, and do not seem to cause undue problems.

* Member, AIAA

Image Size Requirements

The primary size requirements for the target image are set by the combat scenario itself. Image size ranges from an approximate 10° diameter image corresponding to a 50 ft. wingspan target vehicle at about 300 ft. range to a 7 arc-minute diameter image corresponding to that same vehicle at a 24,000 ft. or 4 mile range.

Image Resolution

Image resolution requirements are determined by the necessity to observe the image and determine its characteristics at the appropriate range. At the longer ranges, the limiting factor is the visual acuity of the pilot, himself. Literature searches indicate the normal acuity of the pilot's vision is limited to approximately 1 arc-minute, although some pilots will say that they can do better than this. This limitation set the minimum resolution requirement on the target projection system. The Lockheed systems were designed to give a 1.5 arc-minute per TV line resolution at the 10° diameter condition while the resolution at the smallest diameter was significantly better than the 1 arc-minute requirement.

Image Brightness

The requirement for image brightness of a super-imposed image is determined by the need to provide adequate over-power of the background scene. Since the background brightness for the Lockheed simulation domes was specified to be 2.0 ft. lamberts maximum, the requirement for the target image was set to be 6.0 ft. lamberts maximum, providing a 3 to 1 margin for image highlights. While this margin results in some bleed-through of background highlights through target image darker areas, it does not appear to be a problem.

The problem of background bleed-through was considered to be of much less consequence than the technical problems of matching dynamic positions of the target images and an electronic "hole" in the background scene.

Image Motion/Positional Requirements

Since the location of the target image in a dome simulation is a function of the own-ship maneuvers plus the relative motion of the own-ship and the target vehicle, the dynamic performance of the target projector positional servos are very important. As a general requirement, the own-ship maneuvers were predicated on a Mach 1 speed and a maximum roll rate of 250°/s. The target vehicle speed was also considered to be Mach 1 maximum, while the target attitude variation was not important to the target projector motion requirements. The above dynamic requirements had to be modified to determine actual projector panoramic head requirements based on the head positions within the dome, since the requirements provided only the apparent motion of the image as determined at the pilot's eye-point.

The above relative movements determined the maximum zoom lens dynamic performance as well. The relative speed of the own-ship/target vehicle of Mach 2 can be calculated into the dynamic requirements for the image diameter. This results in a dynamic requirement ranging from .01°/s at the maximum range to 65°/s at the minimum range. As a practical matter, the requirement to provide for a Mach 2 closure rate at a 300 foot range is rather unrealistic and a somewhat lower figure was selected for this requirement.

Positional accuracy was arbitrarily set at 1/2° or 30 arc-minutes as observed by the pilot. In certain areas, such as immediately in front of the aircraft, where image correlation with the HUD display etc. is required, this requirement may not be adequate and fine tuning of the positional accuracy may be required.

Dynamic performance of the target projector system has been defined to match as closely as possible with the dynamic performance of the background image Computer Image Generator (CIG). In the Lockheed case, both domes utilize the General Electric Compuscene IV CIG. This system provides an 80 ms. transport delay with a 60 Hz. field rate on top of the normal simulation system transport lag. Provisions are included to adjust a target system transport lag during integration testing for optimum matching between the target image motion and the background image apparent motion.

Zoom Lens

The installation includes a zoom lens assembly in each panoramic head light path. This zoom provides the necessary adjustment to maintain a constant image diameter throughout the FOV plus the ability to change that diameter as a function of target vehicle range. This zoom ability plus the ability to utilize electronic size control of the image on the projector raster allows range control variations as great as 80 to 1.

Brightness Control

To compensate for the variations in apparent image brightness due to effects of the high screen gain and the changes in projector angles, each projector light path includes an attenuator or iris mechanism to allow control of the light intensity as a function of image location. This mechanism is also used to compensate for differences in brightness due to the variation of image size at the different apparent target ranges.

Target Projector Control

The target projector panoramic heads, zoom lens assemblies, and the other auxiliary functions such as attenuators, shutters, halo diaphragms, etc., are controlled through the use of digital control systems. In Lockheed's first target projector system, this was provided by an integrated control system where the servo control loops were handled by an independent 68030 micro-computer. Other computations were handled within the Gould mini-computer used as an I/O computer in the Lockheed simulation.

These include:

- computation of head azimuth/elevation for each individual panoramic head,
- computation of zoom lens settings as function of individual panoramic head position and desired image FOV,
- control of attenuator/brightness iris as function of individual panoramic head position,
- control of shutter mechanism to hand off image from one panoramic head to another as required at certain image locations, and
- computation of de-roll correction angles to be applied to the image rotational commands given to the computer image generator (CIG) as a function of the individual panoramic head position.

In the Lockheed second target projector system, the addition of two additional target images (a total of four) increased the computational load. This resulted in the decision to provide all of the above computations in addition to the servo loop computations in dedicated micro-computers. Two 80386 micros were used for this application.

De-Roll and Panoramic Head Hand-Off

Use of an azimuth/elevation panoramic head projecting a TV image results in a projector roll of the image as the azimuth axis is rotated. The image computation in the CIG must be corrected for this projector roll. Without this correction, a target image would be perceived as cart-wheeling around the dome if given a pure azimuth motion command. Such a de-roll correction is done for each individual head.

Since, in the Lockheed installation, a single target image from the CIG is used for the two target projector head source Talarias, the de-roll correction must be synchronized with the hand-off from the front panoramic head to the rear head or vice versa. This synchronization requires very close attention to the shutter timing and the transport lag of the CIG. While it has been possible to fine tune this timing to make an almost imperceptible change-over, the more practical approach is to utilize a short black-out period rather than allow a possibility of a momentary de-roll error to be seen. It has been found that the black-out is less disconcerting than the sight of a momentary instantaneous attitude shift of the target vehicle.

While the ability to synchronize the de-roll function precisely with the actual image motion is difficult due to the transport lag of the CIG (approx. 80 msec.), the actual problem seems to be minimal. At high target motion speeds, the eye seems to be incapable of perceiving the precise attitudes. After the motion ceases, the system recovers rapidly enough that the eye and brain tend to ignore the dynamic errors.

Panoramic Head Removal and Replacement

The Lockheed WSSC simulation is designed for use as a development tool for new aircraft design and evaluation. Thus, it was originally specified that the simulation cockpit was to be quickly and easily removed and replaced with a different cockpit. This placed a severe constraint on the target projector system since it was found that the rear pair of panoramic heads would be positioned in such a location that they would prevent the normal roll-out of the cockpit as desired. The mechanical design of the panoramic head optical path structure was made such that it could be quickly and easily removed and replaced without necessity for time consuming re-calibrations and adjustments, thus allowing for the change in cockpits.

Lessons Learned

After going through two development programs of target projectors in short succession, it is apparent that one should have found some areas where the design could have been improved.

The wide dynamic range for the panoramic heads (from essentially 0°/s to over 500°/s and the extreme magnification of the head motion to the projected image makes the servo control extremely important. Any jitter or oscillation of the head becomes immediately noticeable.

While color target images are subjectively more pleasing and desirable, the added brightness of a mono-chromatic light valve projector would have made the optical design less demanding.

Acknowledgements

The author wishes to thank the engineers and technicians from Sogitec Electronic Division and Hughes Support Systems for their inputs into the actual designs of the target projector systems and their forbearance with the author during the development period.

89-3319-CP

THE ENLARGED FILED OF VIEW FIBER OPTIC HELMET MOUNTED DISPLAY

M. Thomas*
The Air Force Human Resources Laboratory
Williams Air Force Base, AZ

B. Barrette**
CAE Electronics Ltd.
St. Laurent, Quebec, Canada

M. Shenker and P. Weissman+
Martin Shenker Optical Design, Inc.
White Plains, NY

Abstract

Four years ago, the development of an Enlarged Field of View Fiber Optic Helmet Mounted Display (EFOV FOHMD) was initiated as Phase V of the FOHMD program. The objective was to dramatically increase the 32 degree per eye field of view (FOV) delivered with the Phase IV FOHMD. The original, rather aggressive, goal was to deliver 100 horizontal degrees per eye. With a 40 degree overlap the resulting instantaneous FOV is approximately 160 degrees horizontally by 83 degrees vertically. During this phase further advances leading to a fully eye-tracked area-of-interest operation have been made. Two oculometers have been developed to the point where eye-servoed operation can be evaluated. Eye servo range has been considerably increased, to plus/minus 45 degrees horizontally by plus/minus 25 degrees vertically. Continued refinements to the optical head tracker have enlarged the tracking envelope to totally encapsulate an F-16 cockpit. This next year will be spent integrating these component technologies, refining system performance, and evaluating the training effectiveness, to provide the Tactical Air Force (TAF)with a cost effective mission simulator.

Introduction

The fiber Optic Helmet Mounted Display (FOHMD) is a state-of-the-art visual display system for tactical aircraft flight training simulation, being developed as a joint US/Canadian effort. A prototype FOHMD, designated the Phase IV system, was delivered in 1985 to the Operations Training Research Facility operated by the Air Force Human Resources Laboratory (AFHRL) at Williams Air Force Base in Arizona. This system has already been described in the open literature (1),(2).

The purpose of the FOHMD is to provide the fighter pilot with a view of the outside world that corresponds to what he would see from an actual aircraft. The displays used in flight simulators for commercial and military transport aircraft present a field of view (FOV) which matches the transport pilots' view of the world as constrained by the cockpit windows. In a fighter, a bubble canopy allows him to see in all directions. To provide this field of regard, the FOHMD places a collimated display, with a limited instantaneous .FOV, directly in front of the pilots' eyes. The appropriate imagery to display is then determined by head tracking the pilot.

Evaluation of the Phase IV system indicated that it had sufficient horizontal FOV, 127 degrees, to support air to ground (A/G) and air to air (A/A) tasks (3). A recent study at HRL, using a dodecahedron based display, indicated that 160 degrees instantaneous FOV, might be an improvement (4). The Phase V development is an attempt to address these questions.

Studies will be performed on both displays to measure pilot performance in a variety of A/A and A/G tasks. Transfer of training experiments are also planned as performance metrics by themselves are not a good indication of the training effectiveness of a particular device (5).

In common with other visual simulation devices, no single parameter can be optimized without degrading others. In general, an FOHMD with a smaller FOV will have lower weight and inertia as well as higher resolution. The planned studies will also consider these and other variables in determining the training effectiveness of the FOHMD system.

---

* Electronics Engineer, Fiber Optic Helmet Mounted Display (FOHMD) program manager

** Manager of Visual Development
Member of AIAA

+ President and Vice President of Martin Shenker Optical Design.

It is unlikely that the temporal fields would cause concern with this vignetting problem. However, 12:00 o'clock vignetting at the nasal fields probably results in a diminished overlap with eye movements. It is not obvious that this is bothersome to a person using the display. Perhaps this is due to the filling of the eye, opposite the vignetting eye, with visual information. More testing of this condition is necessary to answer this question. Through redesign this vignetting can be eliminated at the expense of larger pancake windows or reduced eye relief.

Major System Improvements

The Phase IV FOHMD employed a four camera optical head tracker (OHT) which viewed six infrared LED's arrayed on the top of the helmet, as depicted in Fig 8. The original tracking envelope was a symmetrical box centered on the pilot eye point. In order to optimize the tracking envelope to a modern TAF aircraft, the F-16, the camera orientation and spacings were altered for the Phase V system.

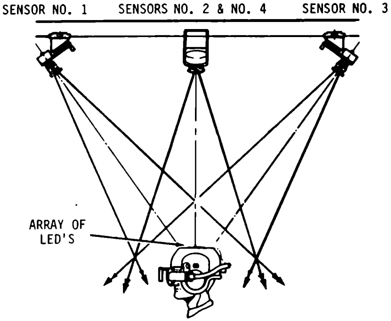


Fig 8 - Optical Head Tracker Configuration

The Phase IV OHT used expensive and fragile accelerometers for lead prediction (6). To reduce costs and increase ab.ility to tolerate impacts, the accelerometers were replaced with rate sensors. Rate sensors have worked acceptably for head motion prediction and are now used on both FOHMD systems. The tracking envelope, accuracy, and motion prediction currently available with the OHT operating at 60 Hertz, are considered adequate to support TAF simulation needs. Table 1 shows the head tracker range and accuracy which resulted from the Phase V developments.

TABLE 1. Phase V Developed Optical Head Tracker Range

Operational Envelope		Resolution
X	: +/- 30 cm	0.025 cm
Y	: +/- 30 cm	0.025 cm
Z	: +/- 20 cm	0.025 cm
Yaw	: +/- 170 deg	0.08 deg
Roll	: +/- 65 deg	0.08 deg
Pitch:	+90,-60 deg	0.08 deg

Color and luminance matching are difficult on the existing 82 degree FOHMD due to the non-uniform characteristics of the light valve pupil and an inadequate amount of diffusion at the output end of the fiber cable. At the present time, neutral density filters are used to match the background brightness with the inset, and the resultant system white field brightness has been measured at 40 footlamberts. That is approximately the brightness of a normally lighted room.

The 100 degree system would have had even greater brightness matching problems with the inset if a different approach hadn't been taken. The standard General Electric light valve has a non-uniform pupil with regard to color, the effect of which has been seen in the 82 degree system inset. If a standard light valve were modified to have a white pupil it would have a significant loss in resolution. Because of this, it was decided to use a multiple light valve (MLV) that was modified to yield a spectrally uniform pupil. With the use of this new light valve, the 100 degree FOHMD is much brighter, and exhibits a much more uniform color range, than the 82 degree FOHMD. Inset resolution has also increased because of the better modulation transfer function (MTF) of the MLV as opposed to the standard single light valve General Electric Talaria Projector. The 82 degree FOHMD will be modified to take advantage of this approach in the near future.

Presently the EFOV FOHMD background channels exhibit a rapid fall off in brightness as the eye nears the edge of the display exit pupil, while the inset has a reasonably uniform exit pupil. The source of this problem is the current light diffusion technique used to create uniform image brightness. Diffusion is accomplished by finely grinding the end of the fiber optic bundle at an imaging surface. The brightness falloff will be corrected as better diffusion techniques are developed.

Investigating better diffusion techniques has lead to another potential improvement. An increase in the system MTF, and inset brightness, appears possible, if the diffusion can be moved from the bundle end, to the relay optics of the background projectors. The fiber optics which feed the imagery to the helmet optics, Fig 9, have been developed to support the FOV demanded by the background, and the resolution required for the AOI (7).

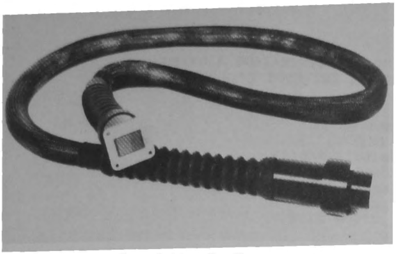


Fig 9 - FOHMD Fiber Optics Bundle

Schott Fiber Optics has recently developed bundles with very regular structure and 5 micron fiber optic core size. These bundles have a limiting resolution of 180 linepairs/mm when polished and 140 linepairs/mm when the fine grind is applied. If diffusion were done in the background only, and polished bundles were used, the limiting factor in system resolution would then become inset size. The use of inset sizes smaller than 25 degrees could then significantly increase system resolution. The limiting factor would once again probably be the fiber optic bundles. The size of the inset will be investigated over the next year. The 82 degree FOHMD, because it magnifies the bundles less than the 100 degree FOHMD, will always be capable of higher resolution and is a more likely candidate for this type of improvement.

Eye Tracking - The Near Future

Eye tracking technology has made considerable progress since the Phase IV FOHMD was developed. At a recent meeting of the Society of Photo Instrumentation Engineers, two oculometers, developed during Phase V, were presented (8). One is a refined ISCAN system, and the other was developed by a University of Toronto group, now incorporated as El-Mar. Figures 10 and 11 illustrate approximately how either of these systems would appear when mounted on the helmet. Both eye trackers use a dark pupil approach with a two dimensional CCD detector array and multiple illuminators. Multiple illuminators provide better lighting and help increase the tracking range. The Phase IV helmet has recently been flown in the eye tracked mode and is currently the testbed for these Phase V developments.

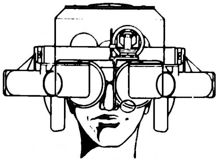


Fig 10 - FOHMD with Eye Tracker

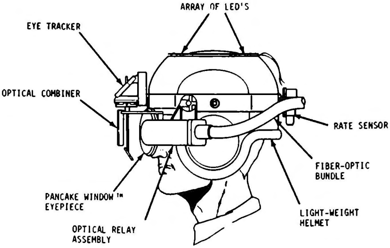


Fig 11 - FOHMD Helmet Details

Considerable refinement will be needed to improve eye-tracker performance and to increase tracking range. However, it appears that a functional oculometer will be developed within a year, based on one or both of these competing technologies. However, the optical performance of the FOHMD is also evolving, thus, generating new demands on the oculometer. If the FOHMD is fitted with a smaller AOI, this will significantly increase the requirement for rapid and accurate oculometer output.

Conclusions

The 100 degree FOHMD has yet to exhibit the level of performance necessary for it to be considered a production system. The technology appears to be in place to finish developing the EFOV FOHMD into a functional simulator. Continued refinements already planned should correct the problems the system currently exhibits. Studies on both these devices will then allow an FOHMD to be specified which will satisfy the Tactical Air Forces training requirement.

Acknowledgements

The research and development work leading to this paper was performed under USAF Contract No. F33615-81-C-0012. As this is a joint cost shared program under the auspices of the US/Canadian Defense Development Sharing agreement, one half of this research was paid for by the Canadian Government.

1. L. Hanson, "Fiber-Optic Helmet-Mounted Display: A Cost-Effective Approach to Full Visual Flight Simulation", Proceedings of the Interservice/Industry Training Equipment Conference, 1983.

2. B. Welch, M. Shenker, "The Fiber-Optic Helmet-Mounted Display", Proceedings of the Image III Conference, PP. 345-361, (1984)

3. R. Kruk, Dr T. Longridge, "Binocular Overlap in a Fiber Optic Helmet Mounted Display", Proceedings of the Image III conference, pp. 363-379, (1984)

4. 1Lt K. Dixon, 2Lt G. Krueger, 2Lt V. Rojas, Dr D. Hubbard, "The Effect of Instantaneous Field of View Size on the Acquisition of Low Level Flight and 30 degree Manual Dive Bombing Tasks", Proceedings of the Society of Photo Instrumentation Engineers (SPIE) Technical Symposium on Aerospace Sensing (TSAS), (Mar 1989).

5. AGARD Advisory Report No.164, Characteristics of Flight Simulator Visual Systems, May, 1981.

6. B. Welch, M. Shenker, The Fiber-Optic Helmet-Mounted Display", Proceedings of the Image III Conference, PP 356,357, (1984)

7. M. Thomas, Dr W. Siegmund, Capt R. Robinson, "Fiber Optic Development for Use on the Fiber Optic Helmet Mounted Display", Proceedings of the SPIE TSAS, (Mar 1989).

8. Capt R. Robinson, M. Thomas, Dr P. Wetzel, "Eye Tracker Development On The Fiber Optic Helmet Mounted Display", Proceedings of the SPIE TSAS, (Mar 1989).

USING A DUAL-EYEPOINT VISUAL SCENE GENERATION SYSTEM TO PRESENT BOTH THE PILOT'S AND GROUND THREAT VIEWS

Frederick B. Fleury
Donald D. Smart
Flight Simulation Laboratory
Fort Worth Division
General Dynamics Corporation
P. O. Box 748 MZ 5939
Fort Worth, Texas 76101

ABSTRACT

Two critical issues of the Close Air Support Study required the Engineering Flight Simulation Laboratory (FSL) at General Dynamics Fort Worth Division to configure a dual-eyepoint visual scene generation system to present both the pilot's and ground threat operator's scenes. The first issue the simulation facility had to solve was how to present the visual cues that would allow the pilot to locate and deliver weapons against an assigned target set. This issue involved the visual scene presentation, the amount of clutter, visual effects, and the pilot's field of view. First, visual scene presentation had to balance best scene resolution against the field of view the projection system provided to the pilot. Second, since a visual scene generation system cannot present real-world clutter, the system had to be tuned to provide the best scene without overloading. Third, visual effects representing battlefield conditions had to be analyzed and incorporated in the scene. Last, with the above parameters driving the requirements, a cockpit station was selected with a projection system whose field of view would accommodate the aircraft maneuver limits.

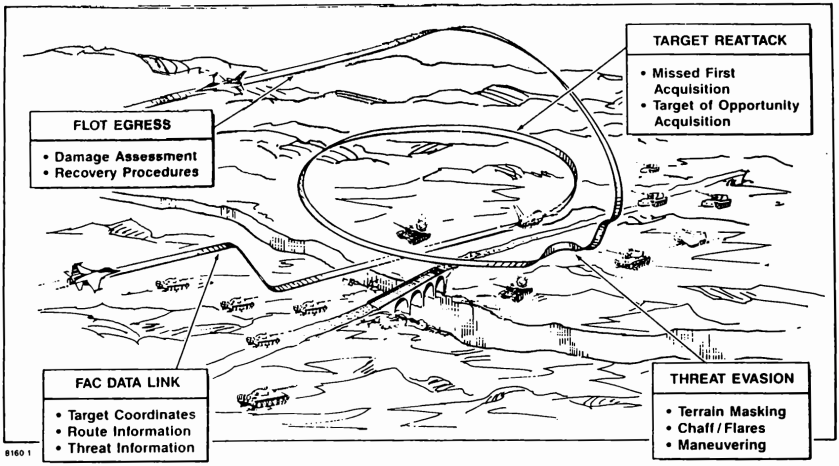
The second issue involved the aircraft survivability against the ground threats located on or near the forward line of troop (FLOT). The study could have used a software model to represent the detection, tracking, and missile launch process of a threat operator. It was opted to use man-in-the-loop as the threat operator to influence the data spread of the survivability numbers. This decision then required the FSL to configure the second eyepoint to present the video for the surface-to-air threat electro-optical tracker.

This paper describes the design considerations for selecting the visual display system, building the visual scene database and optimizing a closed loop response of the E-O tracker video across a fiber-optic data link to the surface-to-air threat simulator.

INTRODUCTION

Figure 1 shows a typical close air support mission. The CAS aircraft loiters in friendly territory until the Forward Air Controller (FAC) supplies the mission target instructions. These instructions include target location, ingress and egress route information and expected threat types in the area. The data can be manually entered into the avionics or automatically entered via data link. This information is used to calculate steering to the target. For the purposes of this study all data were entered into the system via data link. The pilot sets up his ingress to the FLOT area and maneuvers to locate his assigned target. The pilot does a pop-up maneuver and at the peak of the roll-in starts the visual search for the tank line. On detection, the pilot rolls wings level, enters the dive to put the target pipper on the tank. If he does not see the target, then the decision to egress or start a reattack run must be made. The pilot, during his weapon delivery, must detect and evade the surface-to-air threats in the area.

The threat vehicles, on the other side of the FLOT, will be with the tank line or slightly behind. The simulated threat (SA-8 type) will contain both CW illumination and EO TV tracking capability. The threat operators, after radar detection, will attempt to lock the EO tracking periscope on the target as the prelude to launching a missile.



THE CLOSE AIR SUPPORT MISSION
FIGURE 1

These are the two elements of this study. One, how the target cueing accuracy affects the target acquisition and two, in the one-on-one against the passive threat, how the aircraft speed and maneuverability affects survivability.

CONCEPT DEFINITION

Target cueing and survivability were the issues that shaped the study requirements. A concept phase was conducted to answer the questions on the key components needed to accurately represent the environment to the pilot. The high-resolution visual presentation with maximum terrain clutter, realistic target models and battlefield special effects were key factors for acceptable simulator fidelity. The above factors dictated that the scene represent the European terrain and weather norms. Present and future capabilities of target locating systems were considered to assess the question of cueing accuracy. The target cueing was broken down into models representing target locating accuracies between 350 and 3500 feet. Two aircraft models were considered: one was a highly maneuverable but relatively slow concept aircraft; the second aircraft was modeled after the F-16. These models were used to best represent the capabilities of a future CAS aircraft.

This left the issue of how to approach the survivability question. The concept phase used a software model to operate as a surface-to-air threat. It was determined that the programming of this model in the area of detection, identification and tracking was unrealistic. The human element could not be accurately programmed within the limits of this study. The answer was the interfacing of the aircraft simulation to a validated surface-to-air threat simulator. Since today's battlefield threats involve both rf and optical acquisition, a method of providing an accurate video tracking capability had to be developed.

To summarize the above paragraphs, the Engineering Flight Simulator started with the following requirements to configure a simulation that could provide good effectivity numbers on "target cueing" and "survivability".

1. A visual scene that would use all the image generation system capability in the area of throughput rates.

2. A projection system/cockpit station that would provide the best resolution, yet allowed the pilot to maneuver for weapon delivery.

3. A closed-loop link into an offsite threat laboratory that had a large built-in delay in transmission times.

DESIGN CONSIDERATIONS

The first design decision involved the selection of a visual display system to meet the project's requirements. Two different systems were available in the laboratory at this time. One possible system was a 24-foot Dome using three General Electric light valve projectors to produce the image to the pilot's eyepoint at dome center. The resulting presentation had a 163 deg. H x 60 deg. V field of view, with resolution of 5.2 arc min per optical line pair (arcmin/OLP). The second system consisted of three high resolution monitors projected through wide angle collimated lenses set in front of the cockpit. This system produced 138 deg. H x 38 deg. V field of view at a resolution of 2.5 arcmin/OLP. Since target cueing was one of the critical parameters to be measured, the second system was chosen. This choice sacrificed the larger FOV of the first system for the higher resolution visual presentation of the second.

The design of the gaming area was the next decision. A central European database was selected to portray the complex weather and terrain features that might involve a close air support mission. Defense Mapping Agency (DMA) data for a small area of approximately 50NM x 35NM was obtained and processed. The Close Air Support database was produced from elevation data in the Germany Area to be a low-level, very dense, forest terrain. It was produced to run at 50 Hz or 20 millisecond update rate. The CT6* Image Generator required about 17 milliseconds of the maximum 20 to complete each frame. The number of polygons that the CT6 image generator could process was between 8000 and 9000 when integrated with the three WAC window cockpit station. This number of polygons is about 2.5 to 3.0 times what a standard large area database can produce.

The database was designed for a visibility of 30,000 feet. This short visibility number allowed all the polygons to be drawn very near the eyepoint, allowing an extremely high scene density. With the visibility set, the database can also be designed with a 250-meter (830 foot) post spacing with a 50-foot elevation filter. Each 250-meter square is broken up into two right triangles. These two triangles are packed with 13 randomly placed evergreen trees. With this packing efficiency, over 1,000 trees per square mile is achieved on the CT6 image generator. This compares to only 66 trees per square mile on the standard databases with high visibility.

The CAS database also had the ability to place moving targets over the terrain in real time. These included aircraft, missiles, tanks, and dynamic smoke screens. The total capability to the system is to drive up of 12 dynamic targets at one time.

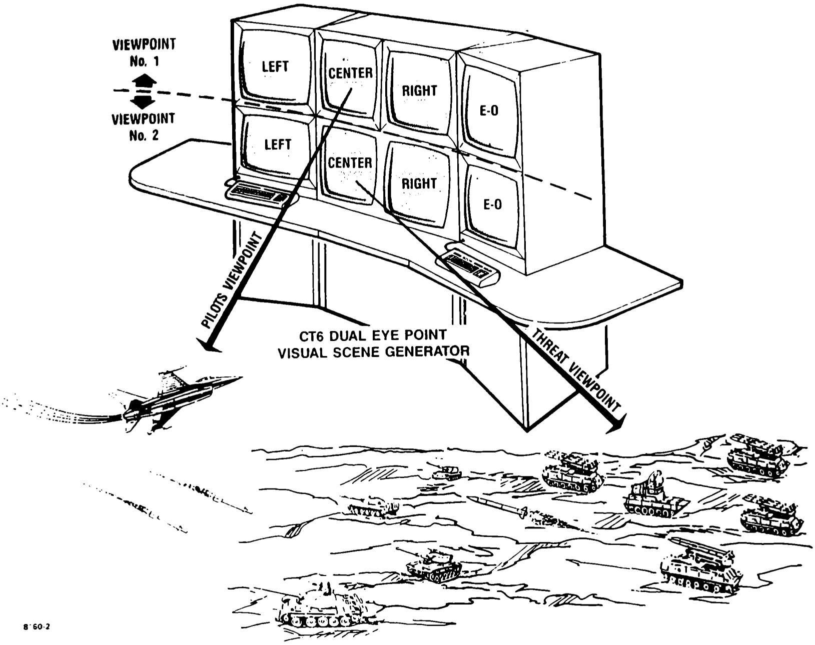
The target cueing models written for this study included current and future navigation systems ranging from a common gyro INS to a more accurate Digital Terrain System (DTS). This DTS tracks the position of the aircraft in a terrain elevation database much like a visual scene generator uses. This position and elevation data are then verified by the RADAR altimeter. The uncertainty of the FAC's position information then had to be inserted into the parameters to arrive at the total target cueing solution.

The aircraft aerodynamic and performance models were baselined to a standard F-16 model. This formed the high-speed end of the study parameters. A second model was developed with increased angle-of-attack and turning performance, while limiting the aircraft speed. This model was used to represent the low-speed, highly maneuverable concept aircraft.

The last design consideration involved the mechanization of the threat operator's visual scene. The Flight Simulation Laboratory had the correct assets for this task. The CT6 is a dual-eyepoint (i.e., dual channel capability) system that can operate as (1) a single-image generator, (2) two separate eyepoints in the same database, or (3) two different simultaneous simulations. Figure 2 is a representation of the control and display console of the CT6 and how the two eyepoints were assigned to the individual elements of the simulation. The center channel of Viewpoint #2 represented the threat operator's sight.

The threat operator's viewpoint video generated by the CT6 was mixed with cross-hair aiming symbology generated by a Sanders-8 raster graphics unit. This

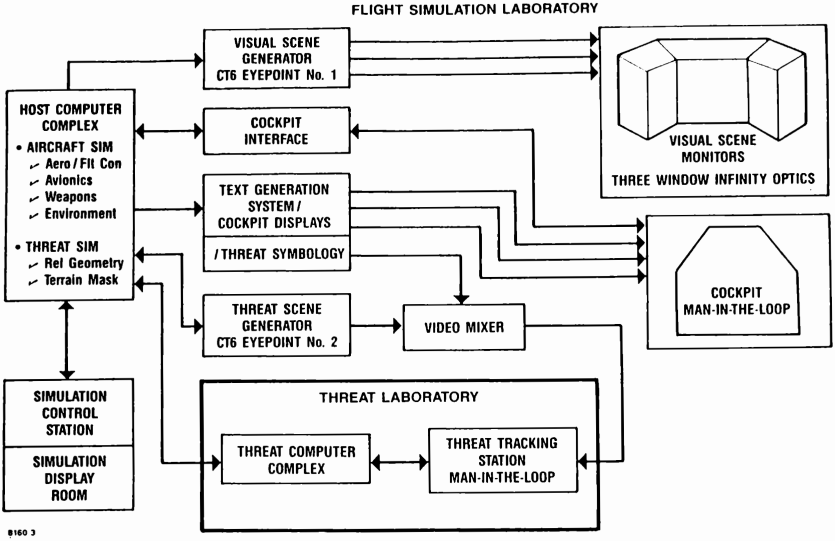
*Model name for Evans & Sutherland system.



EYEPOINT ALLOCATION BETWEEN SIMULATION ELEMENTS
FIGURE 2

mixed video was then encoded for transmission through the fiber-optics link to the threat laboratory. This encoding and decoding of data, video, and audio was performed by the Artel fiber-optic interface on both ends of the fiber-optic link.

Figure 3 shows the configuration of the combined simulations for the Close Air Support study. The aircraft aerodynamics, avionics, weapons and environment were hosted on three supermini computers supported by an array processor. The threat geometry and terrain masking effects were also calculated by this host complex and shipped to the Threat Laboratory. The threat scene generator (CT6 eyepoint #2) provided a video representing the view through an enhanced optical ranger finder. This video was mixed with threat symbology and shipped to the threat laboratory. The display in the threat laboratory was controlled by two handwheels representing elevation and azimuth. These handwheel angles are transmitted to the Flight Simulation Laboratory to position the threat operator's eyepoint. This represented the close-loop operation for an optical tracking system.



SIMULATOR CONFIGURATION
FIGURE 3

TESTING AND VALIDATION

During the simulator testing phase we found the three high-resolution monitor system to have adequate display resolution to present the targets in the visual scene within real-world expectations. The target acquisition ranges were verified by comparison to visual detection range test data. The limited field of view of this type of system prevented the pilots from performing the preferred offset pop-up attack on the targets. Visual scene blurring at fast roll rates was also found to be a problem with this type of configuration due to data throughput times. The visual scene database, built using 800-meter post spacing between the datapoints, was found to provide good terrain detail to depict the Central European area. The tree texture patterns overlayed on the terrain skin seemed to provide the pilot the most effective speed cues in the visual scene. The 5NM visibility modeled into the database allowed the Image Generator to build the high detail in close to the aircraft without the additional load of building the database further than the visibility setting. Color tuning of the database required more development time than expected because of its direct effect on the target acquisition range. This was due to camouflage, and target contrast against the background terrain. The use of typical armored formations modeled on the Dynamic Coordinate Systems (DCS) also provided increased target cueing and identification to the pilot. Gray-scale contrast also played an important role in providing the correct visual acquisition range to the Surface-to-Air Threats optical tracker. The gray-scale of the aircraft was adjusted against the background gray-scale to vary the target aircrafts contrast. This increased or decreased the aircraft's detection range in the optical tracker.

Both navigation models used sinusoidal drift estimation equations varying the magnitude to provide the target cueing errors in the aircraft avionics. Fixed distance errors were then used to represent both accurate and rough estimates of the FAC's location and were added to the navigation solution to achieve the total target cueing solution. A perfect cueing model was also added to provide calibration capability to match the target cueing to the visual scene generator system.

The toughest problem to solve was the transmission delay between the threat operator handwheel inputs to the scene generation system, and back to the display. This loop and the associated delay times are depicted in Figure 4. The throughput time from the Flight Simulation Laboratory's host computers through the fiber-optic link to the Threat Laboratories'computer complex was found to be approximately 1.25 seconds. To solve this problem, time frame counters were put on each computer system and were used to calculate the delay time dynamically from frame to frame. The FSL compared their current time frame count (TFc) with the delayed time frame count (TFd) received from the Surface-to-Air Threat laboratory. The results are the number of frame counts delayed in 20-ms units.

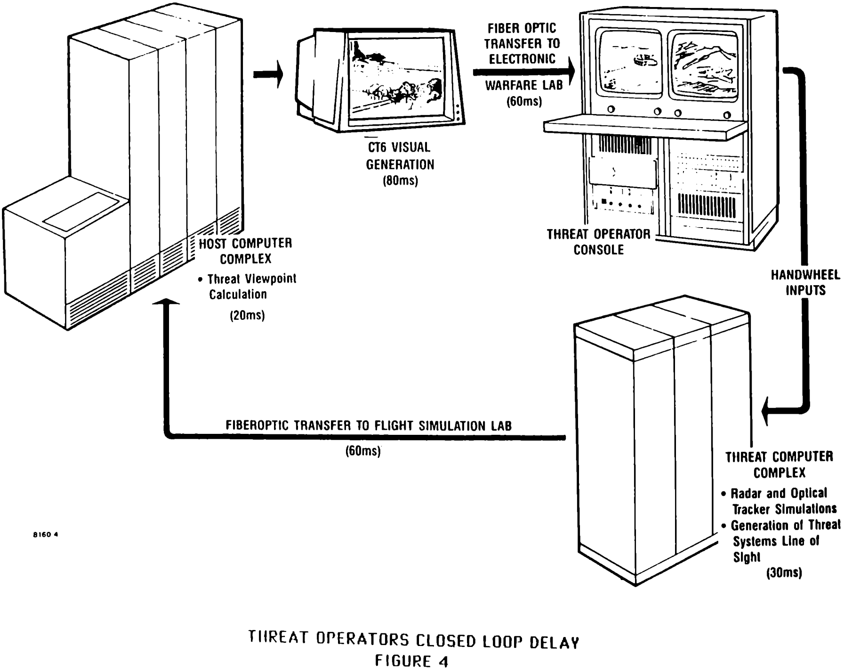
$$N = TFc - TFd \qquad \text{Equation 1.}$$

The FSL then adjusted the Surface-to-Air Threat's pedestal angle positions in the current buffer to compensate for the digital communications lag by the following equations.

$$PEDAZadj = PEDAZ + N * .020 * PEDAZRATE \qquad \text{Equation 2}$$

$$PEDELadj = PEDEL + N * .020 * PEDELRATE \qquad \text{Equation 3}$$

Where: PEDAZ = Pedestal azimuth angle in radians
PEDEL = Pedestal elevation angle in radians
PEDAZRATE = Pedestal azimuth rate in radians/sec.
PEDELRATE = Pedestal elevation rate in radians/sec.


FIBER OPTIC TRANSFER TO ELECTRONIC WARFARE LAB
(60ms)
CT6 VISUAL GENERATION
(80ms)
THREAT OPERATOR CONSOLE
HOST COMPUTER COMPLEX
• Threat Viewpoint Calculation
(20ms)
HANDWHEEL INPUTS
FIBEROPTIC TRANSFER TO FLIGHT SIMULATION LAB
(60ms)
THREAT COMPUTER COMPLEX
• Radar and Optical Tracker Simulations
• Generation of Threat Systems Line of Sight
(30ms)
8160 4

THREAT OPERATORS CLOSED LOOP DELAY
FIGURE 4

PEDELRATE = Pedestal elevation rate in radians/sec.
PEDAZadj = Delay Compensated Pedestal azimuth in radians
PEDELadj = Delay Compensated Pedestal elevation in radians

CONCLUSION

The results of the Close Air Support Study indicated the dual-eyepoint CT6 Image Generation System was a valuable tool in determination of the critical elements of the simulation. In general, the following observations were made.

1. The limited field of view of the projection system was a problem. The pilots had to vary their standard pop-up maneuver to compensate. However, once the baseline runs were established, the study results indicated the expected spread of data on target acquisition.

2. The small visual scene database allowed for maximum amount of manual tuning for object management. The amount of scene detail and special effects provided a more realistic simulation.

3. The use of the CT6 dual eyepoint Image Generation System provided a means of simulating the passive optical threat expected in future battlefields. A controllable zoom capability, terrain masking and high fidelity missile flyouts allowed for the evaluation of the new weapon system against these types of threats in near real-world conditions.

ACKNOWLEDGEMENTS

The authors wish to thank Ms. Toni Vaughn for her secretarial services, Mr. Travis TeSelle for research on the CT6 database, Ms. Cecilia Golden and Ms. Lora Rheuark for editing out our mistakes.

89-3321-CP

NOVOVIEW™ LCV: BALANCING PERFORMANCE AND COST FOR A "LOW COST" VISUAL SYSTEM

Dr. James L. Davis*

Rediffusion Simulation, Ltd.
Crawley, West Sussex, England

Abstract

Aircrew training with flight simulators is accepted as being a valuable supplement to training in the actual aircraft. Enhancing a simulator with an out-of-cockpit visual simulation system further expands this training role, yielding improved training or comparable training at reduced cost. However, a problem exists in providing visual simulation to aircraft users who can't justify typical visual system expense.

The chosen approach examined total system cost rather than component costs, and sought to strike a new balance between cost and performance. Selective capability with flexibility was found to be the key to good performance, while increased standardization was critical for reducing cost. One possible implementation of these findings is Novoview™ LCV, a complete visual system package comprised of computer image generator, generic data base, one of several standardized displays, a visual control console, installation & integration support, and overall product support.

Introduction

Training aircrew in flight simulators, even those without visual systems, is becoming more commonplace owing to well-established cost savings and trainee safety considerations. Despite simulator costs of about $10 million, choosing to use simulator training is relatively easy for major commercial airlines since regulatory agencies such as America's FAA and the UK's CAA allow simulator training to replace time otherwise spent in the actual aircraft. Similarly, it is an easy choice for the military in many cases, either because of the high costs incurred in operating an actual aircraft, or because peacetime flight rules do not permit the range of training needed for combat preparation.

Justifying this level of expenditure is not difficult when dealing with aircraft having a purchase cost of several tens of millions of dollars and an operating cost of several thousand dollars per hour. However, where does this leave the operators of smaller aircraft costing less than one-tenth as much? Certainly they stand to reap the same benefits from training on a visually-equipped flight simulator. One scheme for satisfying the simulation needs of the smaller operators is to provide less expensive simulators possessing less expensive visual systems. However, where does one cut corners in providing the visual aspect of the simulation? The view from the flight deck of a Shorts 360 is not much different from that of a Boeing 747. And the pilot of the smaller aircraft is probably less experienced than the 747 pilot and thus needs proper training that much more.

Scope

To appreciate what "low cost" means in terms of this paper, it's beneficial to first examine the spectrum of simulation devices currently available. Figure 1 graphically illustrates the range of entries in today's simulation marketplace. Plotted are approximate ranges of cost for both the simulator (without visual capability) and a typical visual system appropriate to the device type. These devices range from simple fixed-base procedure and instrument trainers, to motion-base procedure and instrument trainers, to FAA Phase II and Phase III certified flight simulators, to military air combat and weapons tactics simulators/trainers.

As shown, the fixed-based trainers cost about $100,000 and can justify a visual system cost of about $50,000; anything much more or less would not offer the user a balanced capability. This caliber of visual is often satisfied by graphics workstation technology (or its equivalent) running customized flight simulation software. The motion-based trainers typically cost $1-2 million and require a visual system costing about $0.5-1 million. Some companies have sold into this market; however, it is not a thriving market niche. Historically, this can be attributed to an inability to get "good" visual value at this price level. Such is certainly not the case with the well-regulated commercial flight simulation. Simulators costing between $5-10 million usually have visual systems costing several million dollars. This

* Manager, Low Cost Visual Products
Member AIAA

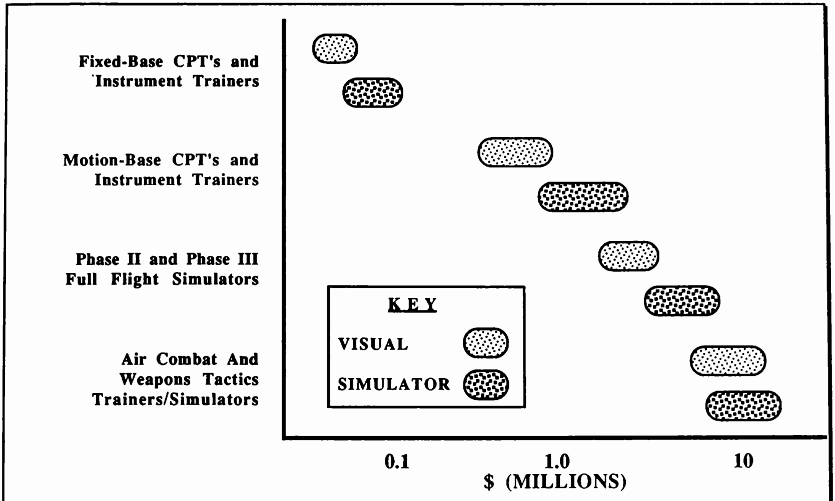


FIGURE 1. SIMULATOR AND VISUAL SYSTEM RELATIVE COSTS AS A FUNCTION OF MARKET SEGMENT

trend holds as well in the military arena, where both simulator and visual system can each cost close to $10 million.

The concern here is not the bottom end of the marketplace, as represented by the left side of Figure 1. Rather, the particular low-cost visual market segment to be addressed is those fixed-base and motion-base cockpit procedure and instrument trainers (costing $1-2 million) that would benefit from a visual package costing $0.5-1 million.

Approach

In defining and developing Novoview LCV, Rediffusion Simulation had as its goal satisfying a new visual cost/performance equation aimed at certain users willing to purchase sophisticated flight trainers, but unable to justify a comparable or larger capital expenditure for a visual system. In the past, this problem was attacked by producing an IG of limited capability (e.g., flat-world, little or no surface texture, digital artifacts) and combining it with a simple data base and collimated or non-collimated display. Often the supplier of the IG or data base was not the same as that for the display. Indeed, often neither was the integrator of the visual system with the simulator; hence, yet another party became involved.

For better or for worse, Rediffusion approached this problem from the standpoint of already being the major supplier of visual systems for the Phase II and Phase III commercial flight simulation market. The standard visual product was and still is customized high performance "complete" visual systems using proven, state-of-the art technology having all the "bells and whistles". Restating the problem from this angle, it is necessary to selectively alter the visual system to reduce cost, but not degrade performance to a point where users feel they are not getting value for money. The previous section pointed out how the visual system is only one of several systems in a flight simulator; the solution to the cost/performance problem begins with a similar decomposition of the visual system.

A visual system is itself comprised of several subsystems, including Image Generator (IG), data base, and display. Because the visual must properly interface with the rest of the flight simulator, mechanical effort is needed to integrate it with the simulator fuselage and motion system (if present) and software effort is needed to integrate it with the host computer and Instructor Operator Station (IOS). Additionally, out-of-cockpit imagery must correlate and be compatible with sensor imagery, cockpit avionics (e.g., HUD), and other perceptual cues (motion, sound, etc.).

A cursory analysis reveals two important keys to the low-cost puzzle. First, care must be exercised to avoid reducing individual component costs at the expense of overall system cost. For example, there is no point in reducing manufacturing cost by relaxing tolerances on the display system if it produces an even larger increase in labor cost owing to complications in assembly and installation. Secondly, standardization must be maximized in order to reduce nonrecurring costs, yet the resulting system must not be so inflexible as to unduly limit its general applicability.

Getting beyond top-level results requires a detailed examination of those visual system attributes that are both performance- and cost-drivers. Such an analysis has been previously described[1]. A summary of this analysis follows.

IG Performance Attributes

It's not unusual for the IG to represent more than 50% of the cost of a visual system. Hence, attributes of the IG play a significant role in any trade-off analysis, and some impact the choice of display system as well. Important attributes include:

Resolution - More pixels imply more video memory, higher video bandwidth, and more high-speed processing for pixel-rendering; all at increased cost. However, decreasing resolution reduces the maximum range at which objects can be discriminated.

Field Of View (FOV) - Larger FOV generally demands more IG channels and displays, brighter display devices, or servo-controlled Area Of Interest (AOI) technology. Reducing FOV saves on cost but limits out-of-cockpit viewing and hence the tasks to be trained.

Scene Content - More surfaces and lights in a scene require more IG processing to yield perspectively-correct imagery. Less scene content makes for simplistic imagery that can limit training effectiveness. Often 2-D texture is incorporated to enhance apparent scene content, the principle being that an expenditure for texture has more benefit than a comparable expenditure for additional surfaces.

Scene Complexity - The IG's ability to handle greater scene complexity relates to supporting more scene occultation. Though most flight tasks don't make heavy demands in this area, periods when the eyepoint is close to a 3-D ground and 3-D objects can increase the need for high scene complexities, as can the use of transparency.

Picture Quality - Good value requires that full use is constantly made of expensive IG processing capability. Additionally, any generated imagery should be free of computational artifacts arising from spatial and temporal aliasing, and jitter. For example, not having anti-aliasing can save on IG cost, but only at the risk of distracting the trainee with unrealistic scene content.

Moving Objects - Moving objects unrelated to the eyepoint (e.g., other aircraft, ground objects, and munitions) permit greater complexity of the training environment. However, aside from impact on IG complexity, moving objects serve as a cost driver since either software or Instructor control must be provided to direct their movements.

Meteorological And Environmental Effects - Because pilots usually fly at all times of day in all types of weather, it is desirable to have a visual system capable of day/dusk/night ambient lighting, limited visibility, reduced ceiling, etc. As with moving objects, flexibility demands control. So aside from increasing the complexity of the IG, variable environmental conditions demand that they be provided for during data base construction and that their control is imparted to the Instructor. These effects also have an impact on display capability and hence cost. For example, daylight simulation (as opposed to dusk/night) generally demands (i) a full color gamut, (ii) more display intensity to yield simulated daylight brightness, and (iii) higher refresh rate to guard against image flicker at the higher brightness.

Update Rate - Increasing update rate means that the displayed scene is recomputed more often for changes in pilot eyepoint location or line-of-sight. It is a cost-driver since rate increases are obtained only from increases in amount or sophistication of computational hardware. Lowering update rate saves money, but the penalty is increased temporal aliasing (manifested as image-stepping or double-imaging). Alternatively, training can be limited to slower eyepoint movement or to reduced FOV's, either tending to reduce the angular rates of objects in the visual scene and thereby making the scene more forgiving of low update rates.

Data Base - Often a simulator user will want training to occur in a world closely resembling that in which the actual aircraft flies. For example, a pilot spending most of his time in the Boston area would probably obtain additional benefit from having a simulator data base depicting Eastern Massachusetts and Logan Airport. The cost of this "custom" data base will be related to (i) quantity of data involved (size of data base) and the difficulty of acquiring source data, (ii) the required fidelity of the data base, (iii) tools available for constructing the data base, and hence (iv) the time (labor) required to build the data base. Efforts exceeding one man-year are not unusual. An alternative to a custom data base is a "generic" data base. If care is taken initially to give it training flexibility, then often it can be provided more cheaply since many users can take advantage of it. The downside, however, is a lack of geo-specific or airport-specific training.

Display Performance Attributes

The display is comprised of (i) a display input device (monitors or projectors), (ii)optical elements (e.g., beamsplitters, mirrors, or screens), and (iii)a support structure. After the IG, the display sub-system shares with installation/integration the distinction of being the next costliest portion of the visual system. The role of the display cannot be overstated; the quality of even the most pristine IG imagery will rise and fall with the presentation made by the display system to the trainee. Display attributes affecting both performance and cost are listed below.

Collimation - There is good reason why collimation has historically found a niche in aircraft flight simulation. First of all, it enhances eye relief (distance from eyepoint to first optical surface) for a FOV and thus often simplifies positioning of the display on the simulator fuselage. Secondly, image size doesn't change with fore and aft head movement, just as in the real world for distant objects). Thirdly, an object's angular position does not change with sideways head movement or, more importantly, with laterally-displaced eyepoints as found in multi-crew cockpits. Finally, the eyes accommodate (focus) and converge (tilt inwards or outwards) as if viewing distant objects, which is comparable to what occurs most of the time from an actual cockpit. Because additional optical components are needed to perform collimation, display costs are higher. Additionally, the additional weight of the optics can have a cost-impact on other parts of the simulator, such as the motion system.

Increased FOV Via Juxtapositioning - A common technique for extending FOV is to mosaic several display channels (which in turn requires several IG channels). To obtain a continuous FOV, each display channel is often blended both in geometry and intensity to ensure continuity between adjacent channels. This blending increases cost owing to the added sophistication needed by the displays. An alternative is simply to abut adjacent channels, leaving a small gap over which no imagery is presented. The argument against abutment is that it's unrealistic and small objects can disappear into the gaps. However, abutment is less expensive both in initial display cost and routine alignment. Also, one can argue that small objects never disappear into gaps for very long owing to the dynamic nature of aircraft imagery.

Field/Frame Extend - The best way to handle a sudden excess in IG scene capacity is to simply extend the period during which that scene can be processed. The impact, as far as the display is concerned, is a longer field or frame time. Most commercial displays offer only a fixed field and frame rate; having an "extend" capability is more costly. However, the alternative to providing for sudden overloads is to ensure that they never occur. This is only accomplished by building sparser data bases and not utilizing the IG to its fullest capacity.

Raster vs. Calligraphy - The calligraphic portrayal of lights yields enhanced realism compared with raster-drawn lights owing to (i) higher brightness, (ii) better contrast relative to the surfaces against which they appear, (iii) finer positioning, and (iv) improved dynamic behavior. However, calligraphy makes severe demands on a display device in terms of power required for linear deflection. In addition, care must be exercised to avoid hysterisis artifacts and improper alignment of raster-drawn surfaces with calligraphically-drawn lights.

Integration Attributes

Though many users of computer image generation equipment simply need a graphics engine and a display for viewing, the user of a flight simulator is not so fortunate. The IG and display must be successfully integrated with the flight simulator in both a hardware and software sense. And, since the simulator is either a training or engineering device, integration must also be accomplished with an Instructor/Operator Station (IOS) to permit control of visual system parameters such as ambient lighting, visibility, and location of moving objects. Some aspects of integration are discussed in more detail below.

Number Of Airport Models - Increasing the number of available airports implies that a means of selection must be provided at the IOS. Additionally, runways need to be aligned with Radio Aids residing in the host computer to ensure correlation of visual with navigation instruments.

Environmental Conditions - Every environmental condition must be controllable from the IOS; this control ranges from a simple on/off to selection from a range of parameters.

Moving Objects - As previously stated, each moving object needs a provision for activation and subsequent control.

Height Above Terrain & Collision Detection - Both of these attributes often require that the IG pass information back to the controlling host computer, thereby complicating an interface that would be unidirectional otherwise.

Weaponry - It's straightforward graphically to depict weapons firing and those effects associated with a hit or miss. However, accurate simulation in a training environment requires computation of trajectory and target correlation with other onboard avionics (e.g., radar, gunsight, or HUD). The implication on integration is added cost.

Display Type - Ease of fitting affects installation cost. For example, the rake of a canopy or the method by which it opens will influence the choice of display. The amount and type of light-tighting can also be a cost-driver. The presence of a motion system implies that greater rigidity with lower weight and inertia are desirable.

Host Computer Interface - The IG demands of the host computer certain information at the commencement of a training exercise, and additional information at regular intervals thereafter. Initially, the host sends information regarding (i) choice of data base, (ii) environmental conditions, and (iii) status of moving objects. Afterwards, the host must communicate to the IG data concerning (i) positional and attitudinal updates of moving objects and (ii) any changes in environmental conditions. The integration required to fulfill this mission must include software to collect and transmit the data residing in the host, and it must include a hardware link capable of communicating the required information at the needed rate.

Novoview LCV™: A Solution

The remainder of this paper addresses one possible approach to the problem of providing a relatively low-cost visual system having good value. Termed Novoview LCV, it is comprised of an IG, data base, display, a visual control console, installation & integration support, and product support (both before and after delivery).Each area will be described in more detail.

Computer Image Generator

The IG selected for Novoview LCV is the ESIG-100 from Evans & Sutherland. Derived from the SP-X family of image generation equipment, it represents proven (over 100 sold to date) state-of-the-art technology with good reliability and low technical risk. In terms of performance, it offers day/dusk/night simulation for up to four channels, both intensity and color-blended texture on surfaces of any orientation, anti-aliasing, up to three simultaneous moving objects, and control of meteorological effects including visibility, RVR, cloud height and thickness, scud, and rain/snow/ice effects.

Because of its importance, special mention will be made of scene management. By way of reminder, scene management refers to efforts taken to ensure that the IG (i) operates close to capacity most of the time, (ii) utilizes this capacity in a manner most beneficial to the pilot (i.e., concentrating detail in the foreground where the pilot can see it), and (iii) in the event of system overload, degrades the image in a graceful rather than abrupt and distracting manner. The ESIG-100 uses the following techniques to guarantee the optimum level of performance:

- **Perspective culling** - Small objects subtending an angle of less than a given threshold are eliminated from further processing by the IG, thus freeing processing capacity for more significant objects.

- **Level-of-detail management** - Objects are modelled in several different ways. Crude models having a small number of polygons are used for representation at medium to large distances; a detailed model having a higher polygon count is normally used for representation when the object is close to the eyepoint. Transitioning between models occurs at a range sufficiently large to minimize observance of a discontinuity. The net effect of this capability is that scene detail is concentrated close to the pilot where his/her acuity can resolve it, and not wasted in the distance where it's not readily discernible anyway.

  In the event that the IG detects an overload situation arising gradually, the transition distances used for switching among crude and detailed models are decreased. This results in the cruder models being used more often and for longer times than designed for originally, but it does reduce IG processing requirements until the overload condition is past.

- **Field/Frame extend** - In the event of a sudden overload (as might occur if a complex moving object enters the FOV of an already complex scene) in which the IG does not have time to execute level-of-detail changes, field/frame extend provides a mechanism for avoiding scene collapse. The time available for processing the image is increased, and the net effect is that the drawing of the new image is delayed and the update/field rate momentarily decreases.

The ESIG-100 differs from the more sophisticated SP-X and ESIG products in several respects. One of these is resolution. Whereas some products have over 700,000 pixels, the ESIG-100 has a bit over 300,000 (yielding roughly 4 arcmin resolution with most display systems). In those applications demanding detection or recognition of small objects at long ranges, this level of resolution will prove inadequate. However, many tasks such as low-level navigation can be satisfactorily trained with this resolution. Furthermore, the anti-aliasing found in the ESIG-100 tends to enhance discernibility of individual small objects!

Update rate is another important differentiator between the ESIG-100 and other systems. Phase II and Phase III visual systems generally update imagery at field rate; i.e., 50 or 60 Hz. A rate of 25 Hz was chosen for Novoview LCV because (i) it allows scene capacities of 500 surfaces/channel and 1000 lights/system and (ii) it compares favorably with motion picture film rates of 24 Hz. In addition, because lower cost visual systems tend to have narrower fields of view than more expensive ones, streaming effects in the peripheral field are less pronounced, thus making temporal aliasing less of an issue.

The ESIG-100 is a pure raster device unlike other products in the SP-X range. Though not a significant cost driver in the IG (it was already designed into it), it would have had a significant impact on display cost. The net effect is to compromise light fidelity; this is especially noticeable at dusk/night. However, light attributes such as (i) straight and curved strings, (ii) random intensity, (iii) horizontal and vertical directionality, and (iv) flashing, blinking, strobing, etc. are still present.

Generic Data Bases

Novoview LCV offers either a generic civil or military data base comprising (i) an airport/airfield, (ii) surrounding 3-D terrain with cultural features out to a radius of 40-50 miles, and (iii) single textured polygon representations of earth and sky that are automatically placed in the visual scene if the trainee travels beyond the boundary of the normal generic data base. To help cater for individual training needs, the airport/airfield model is adjustable by the instructor in terms of runway length, airport lighting, approach lighting, and terminal location.

Display Systems

Modern display systems run the gamut from single-channel direct-viewing to wide-angle collimation to area-of-interest projection slaved to head/eye movement. A low cost visual needs to be more modest in outlook, yet cater to a wide variety of aircraft types. Potential users encompass operators of fixed and rotary wing aircraft having both single- and multi-crew flight decks. Hence, Novoview LCV offers a family of display options in configurations of one to four channels. Monitor-based displays with beamsplitter/mirror collimation are advocated for multi-crew cockpits, whereas front-projection onto an 8' radius spherical screen for direct viewing is recommended for single-seat cockpits or those multi-seat situations where only one crew member is trained at a time. Use of commercial equipment in conjunction with standardized display structures helps to ensure low cost.

Installation & Integration

A standardized Workshare and Interface Control Document (ICD) is part of Novoview LCV and defines the responsibilities of both Rediffusion and the user in terms of installation and integration of the visual system with the simulator. Essentially, Rediffusion installs the equipment and ensures that the host computer can communicate with it (currently via Ethernet); the user is expected to ensure (with Rediffusion's technical advice) that the simulator is mechanically compatible and that the other simulator sub-systems are compatible with the addition of the visual. This permits the use of more standardized display structures than would otherwise be the case. Often this approach represents a turnkey approach for the user. To simplify integration and thereby reduce system cost, Novoview LCV comes with an Instructor's Visual Control Unit that provides control of visual functions in those cases where no capability exists at the IOS.

Product Support

Novoview LCV is provided with both component-level and system-level documentation. After acceptance, the customer is provided with an on-site operations and maintenance course for up to three weeks. And, as long as the equipment is in service, a Customer Service Engineer will pay a one-day visit annually. Optional services are also available at added cost; these include (i) spares, (ii) tools and test equipment, (iii) on-site technical support, (iv) customized data bases, and (v) additional integration tasks, to name just a few.

Conclusion

Striking a balance between visual system performance and cost is very much a function of the type of simulator for which it is intended. Market trends indicate that sophisticated trainers costing about $1-2 million need a visual system costing on the order of $0.5-1 million. Because the bulk of the training market avails itself of more expensive Phase II- and Phase III-compatible visual systems, it was necessary to find a means of trading-off performance and cost to yield a system capable of satisfying this A $1-2 million trainer market.

Looking at the system rather than individual components, it was realized that the key to reducing cost lay in (i) judiciously reducing performance to reduce cost, (ii) implementing standardization to reduce non-recurring costs, and (iii) maintaining system flexibility to ensure economies of scale through broad applicability to a a large user population. Putting these three concepts into practice then required a careful examination of visual system attributes affecting both performance and cost.

One result of this analysis is a product from Rediffusion termed Novoview™ LCV. It is a complete visual system package comprised of (i) an ESIG-100 day/dusk/night high-performance computer image generator, (ii) generic commercial or military data base with adjustable airport, (iii) either collimated or non-collimated display of up to four distinct channels, (iv) Instructor's visual control console, (v) installation and integration support, and (vi) overall product support. A system has already been sold and is due for delivery later this year.

References

1. James L. Davis, "Managing Performance Trade-Offs in Low Cost Visuals", Low Cost Visual Systems Conference sponsored by the Royal Aeronautical Society (23 November 1988) London.

89-3323-CP

EYE MOVEMENT IN AIR-TO-AIR COMBAT TASKS

Capt Kevin W. Dixon
Dr Elizabeth L. Martin
Lt Gretchen M. Krueger
Lt Victoria A. Rojas

Air Force Human Resources Laboratory:
Operational Training Division
Williams AFB Arizona

Abstract

Flight simulators have evolved into complex systems capable of providing training for a number of operational tasks. These systems must make accurate use of the available technology to ensure cost and training effectiveness. Particular emphasis is placed on the requirements of the visual display system to include field of view size, scene content, brightness, and resolution. Research into each of these areas must be accomplished with respect to the desired task to be trained on the simulator.

The current study investigated FOV size as it relates to the visual behavior of pilots performing air-to-air maneuvers in an F-15 simulator. The subjects were eye-tracked and window usage analyzed to determine what portion of the FOV the pilots used during the task and to obtain data of how pilots use their visual system during flight. The Simulator for Air-to-Air Combat (SAAC) proved to be an ideal system because of its large FOV size and high resolution targets. This allowed inferences to be made concerning actual aircraft visual behavior and simulator visual behavior.

Background

Until recently, field-of-view requirements have been investigated by pilot opinion questionnaires and/or direct pilot performance measures with limited emphasis on the requirements for tactical maneuvers. A study by Weikhorst and Vaccaro (1985) investigated the FOV used to perform certain tactical maneuvers. Experienced fighter pilots performed specific portions of air-to- air and air-to-ground maneuvers in the SAAC and the Advanced Simulator for Pilot Training (ASPT). Results indicated that the FOV requirements for air-to-air and air-to-ground maneuvers are different and vary from task to task. This implies that FOV view configurations can be multi-purpose and able to train a variety of tactical maneuvers.[3] The determinations of a multi-purpose field of view size are difficult and interact with cost, training effectiveness and visual system performance (level of scene detail required, resolution).

This difficulty is somewhat overcome when the pilots visual behavior is determined for a number of tasks. The characteristics of visual behavior can be plotted and various plots compared to enhance field of view determinations. In a study performed by Dixon, Martin, Rojas and Hubbard (1988) an eye tracker was employed to test the use of such a device for FOV determinations. In addition to the visual data, direct performance measures and questionnaires data were collected for comparison. The eye tracking system was head-mounted and worn by the pilot while performing the simulator mission. Experienced C-130 pilots flew low level missions with an airdrop on two routes, using two different FOV conditions, (160° H X 35° V and 102° H X 35° V). Performance data showed no strong or consistent effects due to FOV manipulations. However, eye position data revealed an increased use of the front windows and instruments in the limited FOV condition and a decreased use of the window to the left of the pilot. The authors concluded that the peripheral windows may not be required for experienced, pilots, but if present are used, and if absent, alter visual behavior. The change in visual behavior would not have been detected using traditional performance measures. FOV recommendations using visual behavior data may have important training considerations if the pilots are transitioning into the aircraft.[2]

The above study was the first use of the eye tracking technique to assess FOV requirements. The results from the eye tracking data are encouraging and further use of the technique will be made in future FOV experiments on a number of tactical tasks. This paper will present another effort in the evolution of the technique for basic fighter maneuvers performed in the simulator.

Basic fighter maneuvers comprise the tactics involved in visual range fighter combat. The initial training is accomplished in Fighter Lead-in School from various set-ups using a high performance jet aircraft. Further training is done at Replacement Training Units for the specified aircraft. These basic maneuvers are taught by increasing the level of difficulty on the initial set-ups. The training of these maneuvers is done on tactical ranges. If the simulator can effectively train these maneuvers, valuable flight time may be spent on accomplishing complex tasks. The reason for studying eye movement is to describe pilot visual behavior in the simulator and attempt to determine if this behavior is similar to that in the aircraft.

Method

Subjects

Six F-15 instructor pilots, assigned to the 58th Tactical Training Wing, Luke AFB, Arizona, served as subjects. The average total number of hours for these subjects was 1550 with a range of 750-2500.

Apparatus

This research effort was conducted in the Simulator for Air-to-Air Combat (SAAC), located at the 57th Fighter Weapon Wing/Operating Location, at Luke AFB, AZ. The SAAC is used primarily by experienced pilots in the air-to-air combat environment. The SAAC has two fully interactive F-15 cockpits with full instrumentation and weapon systems indicators. The visual system of the SAAC includes eight pentagonal display windows combined with infinity optics within a 16-foot-dodecahedron (See Figure 1).

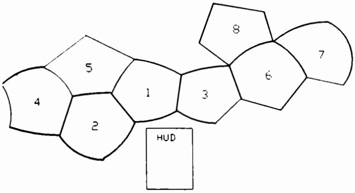


Fig. 1 Simulator for Air-to-Air Combat Window Configuration

A hardware image generation system provides a "checkerboard" ground, sun, sky, and a low-altitude haze layer. The aircraft target image is provided by four closed-circuit television pictures of gimbolled model aircraft displayed on the earth/sky background by means of a small raster inset. The high resolution image allows the target to be seen as far away as three miles. At one mile, the pilot can make out details such as direction, speed, altitude, and wing geometry. Simulator realism includes on-line firing and hit cues, aural effects, g-cues, and weapon sounds.

The SAAC provides an ideal system for FOV evaluations because of its full FOV visual system and tactical aircraft capabilities. The study for this effort investigated a number of tasks that directly relate to the air superiority mission.

Task Description

Each pilot performed basic fighter maneuvers from 4 set-ups. 15 trials were completed for both offensive and defensive set-ups and 10 trials for mutual support and neutral set-up. The total number of trials per subject was 50. All trials ended with a kill, role reversal, or at the maximum time limit (2 minutes). The only weapons available for deployment were aircraft guns in order to better simulate a close-in fighting environment.

The offensive and defensive set-ups were performed at the same initial altitudes and initial airspeeds except the subjects cockpits were switched. An example of these set-ups is depicted in Figure 2. Five trials were performed from each separation point.

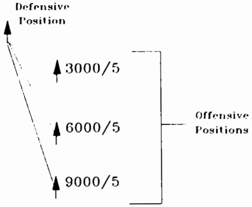


Fig 2 Offensive/Defensive Initial Set-ups

* NOTE: 3000/5 = 3000 FT Separation 5 O'Clock Position

The neutral set-up were performed at an altitude of 19000 ft AGL and an airspeed of 350 knots (See Figure 3). Pilots were instructed to perform five trials of being cleared to maneuver at turn and five trials being cleared to maneuver at pass.

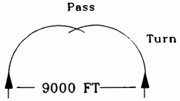


Fig 3 Neutral Initial Set-up

Mutual support set-ups added a third aircraft that was chased by the subjects for five trials (Offensive mutual support) and evaded for five trials (Defensive mutual support). (See Figure 4)

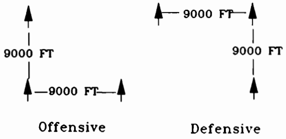


Fig 4 Mutual Support Operations Initial Set-ups

Procedure

Pilots received an initial briefing on the purpose of the study, simulator familiarization, and task guidelines. The experiment was performed over a four day period in which the subjects received one set-up condition per day. Subjects were instructed to perform as realistic as possible with the objective being "to kill or be killed".

Data Analysis

The eye position data was collected using an eye-tracking system consisting of a Model 210 eye movement monitor from Applied Science Laboratories, a video recorder, a time code generator and a computer software analysis program. The monitor employs a photoelectric sensing and processing technique to determine magnitude and direction of eye movements. Infrared illumination is used for eye illumination and sensing with minimal distraction to the subject. The device is attached to a head-band mounted camera which provides a video fixation point capability. The video fixation point capabilities of the device present either cross hairs or a cursor superimposed over a television monitor image of the scene being viewed by the pilot. The image is captured with a video recorder and time coded to complete the data collection procedure. A software program called Tapemaster analyzed the data by coding eye-position relative to previously defined areas of the visual scene. This analysis procedure results in information on the time spent in each window, number of glances per window, percentage of time spent in each window, percentage of glances per window, and percent time per glance.

Results

The results of the eye position data are depicted graphically in Figures 5 - 8. The percent or average number of glances is shown within each of the respective windows. The results for the average number of glances per window is shown in Figure 5. Window 1 has the highest average number of glances and was used most frequently in the neutral set-up followed by the defensive, offensive and mutual set-ups respectively. Window 5 had the second highest average number of glances and was also used most frequently in the neutral position followed by defensive, offensive and mutual set-ups. Window 5 had the next highest average number of glances and was used most frequently in the neutral set up followed by the mutual, offensive, and defensive respectively. Window 2 had the highest use in the offensive, followed by defensive mutual and neutral. Window 2 was used most often in the offensive followed by defensive, neutral and mutual. The Heads Up Display (HUD) and instruments was next in order of average number of glances and was used most often in the defensive position followed by the off Window 3 was used in set ups in the mutual position followed by the offensive, defensive, and neutral positions. Windows 7 and 8 had the lowest average number of glances. Window 7 was used most often in the mutual position followed by the neutral, defensive, and offensive position. Window 8 was used mostly in the defensive position followed by offensive, neutral and mutual.

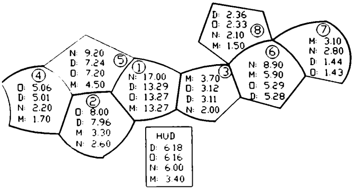


Fig. 5 Average Number of Glances per Window

•NOTE O = Offense. D = Defense. N = Neutral.
M = Mutual Support

The percent of glances per window is shown in Figure 6. The largest percentage of glances occurred in window 1. Window 1 was used most often in the neutral set-up condition followed by mutual, offensive and defensive positions respectively. Window 6 had the second largest percent of glances. The neutral set-up accounted for the largest number of glances followed by the mutual, offensive and defensive positions. The HUD and instruments followed window 6 in the percent of glances per window. The HUD and instruments were required most often in the offensive set-up followed by the defensive, neutral and mutual support positions Window 2 ranked fourth in percent of glances and was used most often in the offensive position followed by the defensive, mutual, and neutral positions respectively. Window 5 followed in percent of glances and was used most often in the neutral set-up position followed by the offensive, defensive, and mutual positions. Windows 4, 7 and 8 had the lowest percentage of glances and were used most often in the mutual support positions.

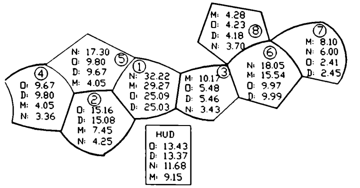


Fig. 6 Percent of Glances per Window

The results of the percent time per window are shown in Figure 7. Window 1 had the highest percent time per window followed by the HUD and instruments. Window 5 had the third highest percent time followed by windows 6, 2, 4 and 7. Windows 3 and 8 had percent time.

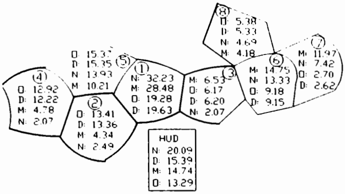


Fig. 7 Percent Time per Window

The results of the average time per glance are shown in Figure 8. The HUD and instruments had glances of the longest duration followed by windows 1, 4, 8, 7, 6 and 2 respectively. Windows 3 and 5 had glances of the least duration.

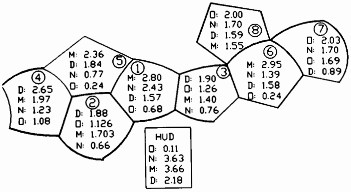


Fig. 8 Average Time Per Glance

Discussion

This study was conducted to investigate the FOV size as it relates to visual behavior of pilots performing air-to-air maneuvers in the F-15 simulator. The objective was to determine what portion of the FOV pilots used during the task and to obtain data on how pilots use their visual system during flight.

Window 1 had the highest average number of glances and the highest percent of glances of any of the other windows. Overall, more time was spent in window 1. This is not surprising since in many of the set-up conditions, window 1 provides information about the environment directly in front of the pilot. There was a general trend in the windows used in specific set-up conditions. It appears that windows 1, 5, and 6 were used most often in the neutral set-up condition. This is the front, upper left and upper right window positions respectively. This is to be expected since a pilot's task in a neutral position set-up is to maneuver into the offensive position which requires use of visual information directly in front and to his immediate right and left.

Conclusions

This study was a preliminary investigation to obtain a general indication of the portion of the FOV being used during specific air-to-air tasks. The data will serve as a baseline for follow-on investigations which will consist of a more detailed analysis of pilot's visual behavior.

References

1. Applied Science Laboratories. Model 210 Eye Movement Monitor: Operation and Service Manual. May 1984.

2. Dixon, K., E. Martin, PhD., V. Rojas, D. Hubbard, PhD. (July 1988). The Effects of Field-of-View on Pilot Performance in the C-130 WST (AFHRL - unpublished). Williams AFB, AZ: Operational Training Division, Air Force Human Resources Laboratory.

3. Wiekhorst, L., F. Vaccaro. (September 1986). Flight Simulator Field of View Utilized in Performing Tactical Maneuvers (AFHRL-TP-86-29). Williams AFB, AZ: Operational Training Division, Air Force Human Resources Laboratory.

89-3324-CP

SIMULATOR NETWORKING IN HELICOPTER AIR-TO-AIR COMBAT TRAINING

James W. Dees, PhD and CPT Timothy R. Cornett
Directorate of Training and Doctrine
United States Army Aviation Center
Fort Rucker, AL 36362

Executive Summary.

This is an unclassified report of a pilot study, i.e., a study in which insufficient data are available to provide statistically conclusive results. All references to weapons performance have been avoided for security reasons. The following summary statements are, therefore, limited to general evaluations of the capabilities of the networked simulators.

According to the aviator participants, the networking of the simulators proved to be a very valuable training experience which could easily be enhanced by better planning if the equipment were more routinely available. The networked AH-1 and AH-64 simulators allowed an evaluation of the effectiveness of various weapons and tactics used in helicopter air-to-air combat (ATAC). This included the ability to obtain comparative hit and kill ratios for each aircraft as a function of range and weapon. It was also possible to evaluate the additional simulator requirements for helicopter ATAC simulation. The field of view of the visual system was insufficient both horizontally and vertically. In close range engagements, one or both simulated aircraft were frequently out of the visual field of the other for extended periods of time. A number of valuable crew comments were provided, as is a list of desired performance measures. The automated weapons scoring capabilities of both simulators were inadequate for testing purposes.

Introduction.

In August 1988, The Singer Corporation demonstrated a relatively simple networking of the AH-64 Combat Mission Simulator (CMS), AH-1 Flight and Weapons Simulator (FWS), and UH-60 Flight Simulator (FS) at Fort Rucker. At the end of this demonstration, three night shifts were made available for a quick research evaluation of this capability. One AH-64 and one AH-1 crew volunteered to participate in an ATAC pilot study involving their respective aircraft. The sample size of this pilot study is inadequate to draw any conclusions with certainty. However, in this instance, there is so little information on the use of simulators in training helicopter ATAC that even the tentative suggestions of a pilot study are of interest.

Procedure.

A total of 45 runs was conducted according to the following scenario:

1. Each run began at 1000, 2500 or 4000 meters distance between the two aircraft with both aircraft at an altitude of 50 meters.

2. Each run began with one of three aircraft orientations:

   a. Head-on

   b. AH-64 advantage, i.e., AH-64 behind the AH-1

   c. AH-1 advantage, i.e., AH-1 behind the AH-64

3. The disadvantaged aircraft, if there was one, would either be instructed to run for cover, then turn and fight after reaching cover, or would be instructed to immediately turn and fight.

4. In 30 runs, each crew had its choice of weapons. However, in order to assure a greater amount of data on the rockets, HELLFIRE and TOW weapons, one of these weapons was dictated for use in each of the 15 remaining runs.

5. If the run was not terminated by a kill or crash, it was terminated at the end of three minutes.

Results.

Only the AH-64 CMS provided suitable data on each firing episode. The AH-1 FWS only provided a summary snapshot for each run. With the exception of these summary data, no AH-1 FWS data were available. Table 1 provides the total number of rounds fired by each AH-64 weapon type as a function of range.

TABLE 1

AH-64 TOTAL ROUNDS FIRED
IN 100 METER INCREMENTS

RANGE	PILOT GUN	CPG GUN	ROCKET	HELLFIRE
20+	0	0	0	22
19-20	0	50	0	0
18-19	0	0	0	2
17-18	0	0	0	0
16-17	0	0	0	0
15-16	0	10	0	1
14-15	0	17	0	0
13-14	0	16	0	1
12-13	0	5	0	1
11-12	85	0	0	1
10-11	0	50	1	0
9-10	0	32	2	0
8-9	38	20	2	1
7-8	15	47	7	0
6-7	84	31	18	0
5-6	45	150	4	0
4-5	214	33	3	0
3-4	236	48	28	0
2-3	245	42	40	0
1-2	147	51	17	0
0-1	138	69	9	0
SUM	1247	671	131	29

Conclusions.

1. The simulation of the gun on the AH-1 included tracers, while the simulation of the gun on the AH-64 did not. According to the gunners, this made the simulated AH-1 gun much more effective.

2. Since neither simulator had an overhead visual system, one crew could fly over the other, then close and kill a blind target. These are artificial conditions. In most actual combat, the existence of surface to air missiles would prohibit ATAC at the altitudes practiced by these crews. Both an overhead visual system and the presence of realistic ground threats are mandatory for the realistic simulation of helicopter ATAC.

3. All four aviator participants stated that the training they received was very valuable. Simply networking two or more attack helicopter simulators together for the purpose of ATAC training will provide a substantial training benefit. However, the following factors should be considered as time and finances permit:

a. The simulators are already very close to full utilization. In order to provide all of our aviators with adequate networked training on our present simulator fleet, additional simulators would be required.

b. A wider field of view visual system would be of great benefit in ATAC simulation. An overhead visual scene is particularly important.

c. A better weapons scoring system is required for most testing. The scoring available on the AH-64 CMS is marginal for test purposes. The scoring system available on the AH-1 FWS is very close to useless. A list of the desired measurements is provided at Appendix B.

d. The crew in each simulator should be able to talk securely, i.e., without their conversation being overheard by the crew in the other simulator as was the case in this investigation.

e. Enough simulators should be connected to allow the use of air-to-air tactics, i.e., there should be a wingman.

f. The aviators in both simulators turned off the motion systems, but expressed a desire to have "g" seats to provide motion onset cues.

APPENDIX A

Crew Comments.

AH-1 Pilot.

1. Crew should be able to talk securely rather than having an open mike where they can be overheard in the other simulator.

2. Pitch and roll constraints should be left out. Mast bumping restriction left in.

3. Should use vectors to assure the opposing aircraft meet each other rather than having an advantaged and disadvantaged aircraft.

4. An overtorque of 120% would crash us. Leave it in the simulator. That's good.

5. The three minute time limit on the run was good, but should fight it out.

6. Concerning cockpit fatigue, you are tired during the graveyard shift, but that is a realistic combat condition.

7. You need a wingman.

8. With a load of 2 TOW, 500 20 mm and 12 rockets plus full fuel, weight and balance were not considered.

9. This exercise on the simulator teaches patience. You must be within range fcr the 20 mm to be effective.

10. The limited field of view behind, below, and above was a real handicap, but the training was great.

AH-1 Copilot Gunner.

1. It was a good learning experience.

2. The experience level between the AH-64 and AH-1 crews was very different.

3. A crash in the cobra occurs with a high angle of bank. A "g" load criterion should be used instead.

4. You need a top canopy to be able to follow in steep banking maneuvers.

5. Simulators don't show the incoming rounds. They should.

6. Tracers are needed on both aircraft simulators. They helped on the AH-1 FWS.

AH-64 Pilot.

1. The scenarios used in the study would be a good introduction to acquisition and recognition training.

2. This is basic air-to-air training.

3. You should go into more advanced scenarios with terrain, mission, and orders, and see what the trainees do.

4. To be a superb air-to-air trainer, you need better visuals and a "g" seat. A wider field of view is required. Better detail would help, but is not critical.

5. How do you deploy to acquire, i.e., do you conceal yourself and go into an overwatch? This should be investigated.

6. A better data recording capability is needed.

7. You need dedicated crews. The tactical expertise was poor in this study.

8. You need isolated communications within crews.

AH-64 Copilot Gunner.

1. Test results are not valid because of crew inexperience in the first part of the test. Late Friday night, the data was fairly good.

2. No planned setup and ability evaluation.

3. For valid data, 4000 meters should have been first, then moved in closer. The three minute scenarios were good.

4. Communications within the crew must be isolated.

APPENDIX B

Suggested Measures In Addition To Those Available Now On The AH-64 CMS

1. Time to first hit by each aircraft

2. Time to kill by each aircraft

3. If there is an advantaged aircraft, the number of times the advantaged aircraft is killed before his adversary is killed

4. Number and type of ordnance expended by each adversary at the time of each of the above criteria

5. Type of ordnance which resulted in each hit

6. Type of ordnance which resulted in each kill

7. Altitude difference between firing and target aircraft at time of hit or miss by missile, rocket, or gun burst (Firing aircraft higher will be positive.)

8. Slant range between firing and target aircraft at time of hit or miss by missile, rocket, or gun burst

9. Difference between angular velocity of target aircraft and munition at time of hit or miss (Gun fire will be considered in bursts. Greater target velocity will be positive.)

10. Difference between the angular acceleration of target aircraft and munition at time of hit or miss (Gun fire will be considered in bursts. Greater target acceleration will be positive.)

89-3325-CP

Recovery Strategies for Microburst Encounters Using Reactive and Forward-Look Wind Shear Detection

David A. Hinton*
NASA Langley Research Center
Hampton, Virginia

Abstract

An effort is in progress by the NASA, FAA, and industry to reduce the threat of convective microburst wind shear phenomena to aircraft. This paper describes an effort to quantify the benefits of forward-look sensing and to develop and test a candidate set of strategies for recovery from inadvertent microburst encounters during the landing approach. Candidate strategies were developed and evaluated using a batch simulation consisting of a point-mass performance model of a transport-category airplane flying through an analytical microburst model. The candidate strategies were then evaluated in piloted simulations using a full dynamic airplane model. The results of this effort indicate that the factor which most strongly affects a microburst recovery is the time at which the recovery is initiated. Improving the alert time by 5 seconds generally provided a greater recovery performance increase than could be achieved by changing the recovery strategy. Forward-look alerts given 10 seconds prior to microburst entry permitted recoveries to be made with negligible altitude loss. This trend was substantiated in piloted tests.

Nomenclature

D	Airplane drag
E_h	Airplane energy height
F	Wind shear hazard index
g	Gravitational acceleration
h	Airplane altitude
Href	Reference altitude for recovery strategies
T	Total airplane thrust
V	Airplane airspeed
W	Airplane weight
W_h	Vertical wind component, updraft positive
W_x	Horizontal wind component, along the airplane ground track
γ_p	Airplane potential flight path angle

Introduction

Numerous air carrier accidents and incidents have resulted from inadvertent encounters with the atmospheric wind shear associated with "microburst" phenomena, in some cases resulting in heavy loss of life. A microburst is a strong, localized downdraft that strikes the ground, producing winds that diverge radially from the impact point. An airplane penetrating the center of a symmetric microburst will initially encounter an increasing headwind, followed by a strong downdraft and rapidly increasing tailwind. The effects of the downdraft and increasing tailwind may easily exceed the climb and acceleration capabilities of the airplane, causing an unavoidable loss of altitude and airspeed. These encounters continue to occur since the ability to reliably predict or detect a microburst in an airplane's flight path, in an operational environment, does not yet exist.

The National Aeronautics and Space Administration (NASA) and the Federal Aviation Administration (FAA) are addressing this hazard through the Integrated Wind Shear Program. The goal is to reduce the hazard of low level wind shear to aircraft

*Aerospace Technologist, Vehicle Operations Research Branch operations through improved airborne and ground-based wind shear detection systems, crew alerting and flight guidance systems, and training and operating procedures. NASA is investigating the airborne aspects of the problem through hazard characterization, sensor technology, and flight management systems.

Previous research has shown the performance available from an airplane following an optimal trajectory (refs. 1 and 2) when full knowledge of the microburst flow field is known. The application of these optimal recovery concepts to practical recovery guidance laws was studied (ref. 3) and the performance of those guidance laws in piloted operations, in the takeoff encounter case, was evaluated in the simulator study described in reference 4. The studies described in references 3 and 4 showed that advanced guidance laws enabled recovery to take place at higher minimum altitudes than with baseline constant pitch techniques, but that recovery altitude was very sensitive to small deviations in airplane pitch history. In piloted simulation tests, the performance differences between various recovery strategies were statistically insignificant. This result emphasizes the need for microburst avoidance.

Two levels of microburst avoidance are possible: (1) totally avoid an encounter with the microburst phenomena, and (2) avoid placing an aircraft in a critical low-energy situation, by initiating a recovery procedure prior to microburst entry. The first level of avoidance is the ultimate goal of the program, but requires a higher degree of sensor development than the second level. Advances in forward-looking wind shear sensor technologies have raised the issues of how much forward-look distance is necessary to ensure airplane survival during a recovery procedure and how recovery guidance concepts will be affected by forward-look data. This paper describes an effort aimed at quantifying the benefits of relatively short range forward-look sensing and developing recovery guidance concepts, for the approach-to-landing case wind shear encounter, utilizing both reactive-only and forward-look wind shear detection.

Airplane Energy Height Analysis

In an effort to determine the amount of forward-look detection necessary to ensure airplane survival, an analysis of airplane energy height during a microburst encounter was conducted. The purpose of the analysis was to gain insight of how changes in microburst strength and detection delays or advances (forward-look) would affect airplane survivability, by examining the changes in energy height across the events. Airplane energy height is defined as:

$$E_h = \frac{V^2}{2g} + h \qquad (1)$$

where V is airspeed, g is gravitational acceleration, and h is altitude. From reference 3, the potential flight path angle of an airplane in the presence of wind shear can be approximated by:

$$\gamma_p = \frac{T - D}{W} - F \qquad (2)$$

where T is airplane thrust, D is airplane drag, W is airplane weight, and F is the "F-factor":

$$F = \frac{\dot{W}_x}{g} - \frac{W_h}{V} \qquad (3)$$

The two wind terms describe the wind shear impact on the climb angle capability of the airplane, in terms of the horizontal shear $\dot{W}_x$ and vertical wind (W_h). Multiplying the potential flight path angle by airspeed provides an approximation to the potential rate of change of energy height.

$$\dot{E}_h = V\left(\frac{T-D}{W} - F\right) \qquad (4)$$

Equation 4 can be integrated across a wind shear scenario to determine the change in energy height. These equations alone do not incorporate the effects of different airplane trajectories through a shear or predict the actual altitude of an airplane after a shear, but can be used to estimate the change in energy height given an event encounter.

Batch Simulation

A batch simulation of various wind shear encounters was conducted to further evaluate the benefits of forward-look detection and develop microburst recovery strategies. The batch simulation consisted of a point-mass airplane model, an analytical microburst model, and a simple wind shear detection scheme.

Airplane Model

The airplane model is based on a Boeing 737-100 flying in a vertical plane. The gross weight was set at 90,000 pounds and sea level standard atmospheric conditions were assumed. During approaches, the wing flaps were set to 25 degrees, the gear was down, and the thrust was controlled by an autothrottle to maintain the 137 knot reference speed. The glide slope was maintained until activation of the wind shear alert, at which time the recovery strategy was initiated. During escape maneuvers, the wing flaps and landing gear positions were left unchanged and the thrust was increased to 24,000 pounds. This emulates the recovery procedures currently being taught to airline crews.

Microburst Model

The microburst model (ref. 5) represents an axisymmetric stagnation point flow that satisfies mass continuity and includes boundary layer effects near the ground. The boundary layer effects and spatial variation in outflow and downflow closely match real-world observations. The microburst has a maximum outflow of 37 knots at an altitude of 120 feet and at a radius of 2,391 feet. The severity of the modeled shear is representative of microbursts that have caused aircraft accidents. As a consequence of the boundary layer effects, the apparent severity of the shear is dependent on the altitude of the encounter.

Wind Shear Detection

Wind shear detection logic was used to activate the recovery control laws. The detection was based on the F-factor of the wind shear. A F-factor threshold of 0.15 was used to determine when the shear had been entered. The threshold F-factor does not occur at any given spatial location in the microburst model, but depends on airplane altitude, flight path, and airspeed. Variable time advance and delay was implemented to simulate forward-look sensors and reactive device delays. The advance is defined as the number of seconds that the alert is given before the threshold F-factor would have been exceeded if no alert had been given. The delay is defined as the number of seconds that the alert is given after the threshold F-factor is exceeded. A delay of 5 seconds is considered to approximate the response of realizable reactive detection systems. The recovery control laws use the two detection discrete signals to begin the recovery strategy and to determine control law gains that are dependent on the alert type.

Candidate Wind Shear Recovery Strategies

Of the various recovery strategies tested, three are discussed below. The control law for each strategy limited the target pitch attitude to the value that would place the airplane at the stick shaker angle of attack and the pitch rate was limited to 3 degrees per second.

Manual Strategy. This strategy is currently in use by the airline industry (ref. 6). Since this strategy was designed to be manually flown in the absence of guidance commands, the exact procedure used will vary slightly from pilot to pilot. For this effort the manual recovery was approximated by initially rotating the airplane to a pitch attitude of 15 degrees. This pitch attitude was maintained if it produced a zero or positive flight path angle. If 15 degrees of pitch was insufficient to maintain level flight, however, the control law would further increase pitch in an attempt to maintain level flight.

Flight Path Angle Strategy. This strategy required the airplane to fly a flight path angle that was a function of altitude, wind shear F-factor, and available airplane performance. When the potential flight path angle was positive, that climb gradient was maintained. When the potential flight path angle was negative, the target climb gradient was altitude dependent. Below a reference altitude (Href) the strategy attempted to climb regardless of wind shear strength, under the assumption that obstacles must be cleared. The target flight path angle was 0.03 radians at ground level, reducing linearly to level flight at Href. Above Href, the strategy maintained one-half the potential flight path angle. This permitted a descent to be maintained at the higher altitudes in order to reduce the rate at which airspeed was lost. Href was set to 100 feet for reactive alert recoveries and 400 feet for forward-look alert recoveries. These altitudes provided the best overall performance in preliminary data runs. Both the positive and negative flight path angle target values were limited to 0.06 radians to prevent excessive climb or descent rates during the recovery. The flight path angle was further limited to prevent descent below the glide slope.

Glide Slope Strategy. This strategy attempted to emulate the characteristics of the optimal approach abort trajectories described in reference 7. That effort showed that the optimal recovery trajectory initially produced a descent and later transitioned to level flight. The level flight path was flown until exiting the wind shear. For this study, that trajectory was approximated by initially tracking the glide slope, at go-around thrust, until the altitude reached a reference altitude (Href). The value of Href was 100 feet for reactive alert recoveries and 500 feet for forward-look alert cases. After reaching Href, the strategy attempted to maintain a zero flight path angle until exiting the shear. The glide slope was chosen as the descent angle to provide obstacle clearance during the recovery.

Real-Time Simulation

Simulator Hardware

The Langley six-degree-of-freedom visual motion simulator was utilized in the piloted tests. The simulator provides a generic transport airplane flight deck equipped with conventional electromechanical instrumentation. The pilot was provided with an Attitude-Director Indicator (ADI), airspeed indicator, turn and slip indicator, barometric altimeter, radar altimeter, vertical speed indicator, heading indicator, and engine pressure ratio instruments.

The recovery guidance was presented on the ADI by conventional dual cue command bars. The recovery guidance was presented to the pilot in the same format for all three control laws. A pitch target was calculated to accomplish the strategy goal, and the pitch command bar was positioned at that point on the ADI. The roll command bar was driven by bank angle, so that the command was nulled whenever the wings were level. Full scale deflection of the roll bar occurred at a bank angle of 60 degrees. A fast/slow indicator in the ADI display was driven by speed error, with full scale deflection occurring at 10 knots of error.

An out-the-window display of terrain was provided on the cockpit forward windows. This visual display was driven by a terrain model board and permitted the execution of visual approaches and landings. No visual meteorological cues or effects were shown. Audio cues for wind and engine noise were also provided.

Pilot control input was through a wheel and control column hydraulically loaded in pitch and roll, hydraulically loaded rudder pedals, and independent throttle levers. A stick shaker function was implemented on the control column.

Airplane Model

The simulator was driven with a full nonlinear math model of a Boeing 737-100 airplane with Pratt & Whitney JT8D-7 engines. The model included lift and drag coefficient data to 24 degrees angle of attack, and the effects on those coefficients of pitch rate, control deflection, and ground effect. Variations in aileron and elevator control forces with airspeed and trim position were fed back to the pilot. The aircraft was flown in the landing configuration at a gross weight of 90,000 pounds. Standard sea level temperature and air density were selected.

Microburst Models

Two microburst models were utilized in the piloted simulation. The first was the same analytical model used for the batch simulation effort. The second model was a numeric representation of a microburst as generated by the Terminal Area Simulation System (TASS) program (refs. 8 and 9). The TASS program implements a numeric atmospheric simulation, which determines parameters such as temperature, wind components, pressure, and liquid water content at grid points within a cube of airspace during the evolution of a microburst. For this effort, the TASS model was initialized with the atmospheric conditions that produced a microburst at the Dallas-Fort Worth (DFW) Airport in August, 1985. The resulting TASS output closely resembles the microburst involved in the DFW accident.

The two microburst models differ primarily in the scale of the event and in the precursors entering the microburst. The TASS model is a large scale event, with the performance-decreasing wind change taking place over a distance of about 12,000 feet, or more than double the corresponding distance in the analytical model. Prior to entering the performance-decreasing region, both models include a performance-increasing region. This region involves only an increasing headwind in the analytical model. In the TASS model, the performance-increasing region involves both an increasing headwind and a strong updraft.

Wind Shear Detection

Both reactive and forward-look detection capabilities were used. The F-factor was used in the implementation of both detection types. In the reactive detection case, knowledge of the local horizontal and vertical wind was used to calculate the F-factor. The reactive alert was given when the calculated F-factor exceeded a threshold value for 5 seconds. In the forward-look detection case, the F-factor was estimated using wind divergence information at a detection point ahead of the airplane. The detection point was located along the instantaneous flight path vector of the airplane, at a distance corresponding to 10 seconds of flight time. The detection threshold value of the F-factor was set at 0.12 for the reactive alert. To compensate for known error sources in the forward-look F-factor estimator and to achieve a forward alert 10 to 15 seconds in advance of the reactive alert, the forward-look detection threshold was set at 0.10.

Alerts were presented both visually and aurally. A red annunciator light directly above the ADI and a warbling tone indicated a reactive alert. The red light was canceled after the F-factor decreased to 0.05 and the audio tone was canceled after three seconds. An amber light above the right corner of the ADI and a steady tone indicated a forward-look alert. The amber light was canceled either by the activation of the reactive alert, or the forward-look F-factor decreasing to 0.05. The audio tone was canceled after 10 seconds, after pilot activation of the escape guidance, or after receiving a reactive alert, whichever occurred first. Prior to receiving an alert, the flight director was used in an instrument landing system tracking mode. Upon receiving a forward-look alert, the escape guidance was activated by the pilot pressing a throttle-mounted Takeoff-Go Around (TOGA) switch. Upon receiving a reactive alert, the flight director automatically switched to the escape guidance.

Experimental Conditions

Three microburst scenarios were used for the statistical analysis. In each case the microburst was located on the extended runway centerline. One scenario positioned the core of the analytical microburst 1,640 feet from the runway. A second scenario positioned the same microburst at an increased distance of 3,600 feet from the runway. The close-in location approximated the microburst geometry of a batch simulation run with an initial altitude of 140 feet. The more distant location approximated a batch simulation run with an initial altitude of 240 feet. The third scenario positioned the TASS model 1,640 feet from the runway. Additional microburst geometries and locations were included in the test matrix to provide variability to the pilots, although too few repetitions were run to be included in the analysis.

Prior to each data run the aircraft was initialized at an altitude of 1,200 feet, on the centerlines of an Instrument Landing System (ILS) localizer and glide slope. The airplane was trimmed for a three degree descent at an airspeed of 137 knots, with landing flaps and gear position selected. The pilot task was to track the ILS until receiving either a reactive or a forward-look wind shear alert. At that point the task was to apply maximum rated thrust and initiate an escape maneuver, using the flight director guidance. Since the performance effect of the alert timing was being evaluated, the pilots were asked not to begin a go around maneuver before receiving an alert. The run was terminated upon ground contact or successful exit from the microburst.

A total of seven research and airline pilots participated in the study. The average amount of turbojet experience was over 5000 hours. Five of the seven pilots had completed the FAA wind shear training aid program. The matrix used for data analysis consisted of three repetitions of three microburst scenarios, three recovery strategies, and two alert times (reactive only and forward look) for a total of 54 runs.

Another 11 runs, not included in the analysis, were inserted throughout the matrix for the alternate microburst scenarios.

Results

Energy Height Analysis

In the results to be presented, data for a Boeing 737-100 airplane were used. The configuration was assumed to be gear down, flaps set at 25 degrees, and a weight of 90,000 pounds. In this configuration the reference approach speed is 137 knots and the stick shaker speed, at go-around thrust, was estimated to be 107 knots. The value of (T - D)/W was set at -0.05 for the approach and at 0.16 for the recovery.

The energy height analysis was conducted for various wind shear strengths and alert time values. The F-factor of the shears was varied from 0.10 to 0.30 and the shear width was 5,000 feet. Alert time was varied from -20 seconds to 60 seconds. A negative alert time indicates an alert received after entering the shear, which simulates a reactive system or pilot recognition of the shear, and a positive alert time indicates an alert received prior to shear entry, to simulate a forward-look device. The results are depicted in figure 1. The figure shows that with a 10 to 15 second delay in detecting the presence of the wind shear, even a relatively weak shear will result in the loss of 300 to 500 feet of energy height.

The benefit of reducing the time required to detect a wind shear can easily be seen. For each second of improvement in the time required to give a reactive alert, the energy height loss across the event is reduced about 40 feet. Still greater benefits are achievable with the use of forward-look wind shear detection and alerting. Alerts given only 15 to 20 seconds prior to shear entry would permit recovery from relatively strong shears with essentially no altitude loss.

Batch Simulation Recovery Altitude Performance

In each batch run, the airplane was initialized on a three degree glide slope, 4,000 feet from the center of the microburst. The geometry of the microburst encounter was varied by changing the initial altitude of the airplane. The initial altitude, wind shear encounter altitude, recovery altitude, and alert time for these runs are shown in table 1. The encounter altitude is the altitude at which the wind shear was detected and the recovery initiated, and the recovery altitude is the lowest altitude during the encounter. The initial altitudes varied from 300 feet to 700 feet above ground level in 100 foot increments and the alert time varied from a 10 second forward look to a 10 second delay in 5 second increments.

Of the factors explored, the factor that produced the greatest improvement in recovery altitude was the alert time. The improvement in recovery altitude with each 5 seconds improvement in the alert time was generally greater than the difference in performance between recovery strategies. Depending on the initial altitude, the recovery altitude increase ranged from 114 feet to 498 feet when the alert time was improved from -5 to +10 seconds. Figure 2 illustrates the relative effect of changing the alert timing or the recovery strategy. The three strategies are shown with a 10 second forward-look alert and a 5 second delay insitu alert. The difference in recovery altitudes between the recovery strategies are small compared to the effect of improving the alert.

Real-Time Simulation

A three factor Analysis of Variance (ANOVA) was conducted on the recovery altitude data to detect significant differences in the three recovery strategies, two alert types, and three microburst scenarios. The average recovery altitudes for the various combinations of the three factors are shown in table 2 and the ANOVA results are shown in table 3. The ANOVA indicates a difference in all three factors, at a 0.01 level of significance. Examination of table 2 shows, however, that the difference in performance between the three recovery strategies was about 20 feet while the difference in performance between the alert types was about 300 feet. Hence, the recovery altitude improvement produced by improving the alert time by 15 seconds was over an order of magnitude greater than what was achieved by changing the recovery strategy.

The statistics also show interaction between the alert and shear and between the alert and guidance at a 0.01 level of significance. Table 2 shows that the alert type produced a smaller change in recovery height with the TASS model than with the analytical model. The data also shows that varying the guidance produced larger changes in recovery performance in the reactive alert case than in the forward-look alert case.

Figure 3 shows examples of aircraft trajectories during runs with the analytical model positioned 1,600 feet from the runway. This geometry, the alert timing, and the recovery strategies are the same as the batch runs shown in figure 2. The characteristics of the recoveries agree well between the batch and real-time simulations. In each case, the manual strategy initially produces a climb, followed by a descent at the stick shaker angle of attack. The other strategies produce a more level trajectory that recover at essentially the same altitude. The recoveries suggest that recovery altitude is not the only parameter of interest in evaluating recovery strategies. The strategies that maintained a lower trajectory produced higher minimum airspeeds and less or no time at the stick shaker angle of attack, while producing similar recovery altitudes as the manual strategy.

The pilots completed questionnaires and provided comments on the alerting and guidance. Of the seven pilots, six believed that the timeliness of the forward-look alert was about right, while one believed that the alert was too late. No one believed the alert was too early. When asked if the 10-second advance alert was sufficient to make a normal go-around procedure more appropriate than a programmed escape maneuver, six of the pilots said yes. The pilots were also asked if airplane configuration should be held constant during escape maneuvers with the two alert types. In the case of reactive-only alerting, six of the pilots indicated the configuration should be constant. In the case of forward-look alerting, only two of the pilots believed it was appropriate to maintain a fixed configuration. The pilots were asked to rate the three recovery strategies in order of preference, with "1" assigned to the most preferred strategy. The averages of all the pilots was 1.3 for the flight path angle strategy, 2.1 for the manual strategy, and 2.6 for the glide slope strategy.

Conclusion

Analytical analysis, batch simulations, and piloted simulations indicate that the factor which most strongly effects a microburst recovery is the time at which the recovery is initiated. In nearly all microburst situations evaluated in batch simulations, improving the alert time by 5 seconds provided a greater recovery performance increase than could be achieved by changing the recovery strategy. Forward-look alerts given 10 seconds prior to microburst entry permitted recoveries to be made with negligible altitude loss. In piloted simulations, the average recovery altitude only varied about 20 feet between the recovery strategies tested. In contrast, the average recovery altitude varied nearly 300 feet between the two alert times tested (-5 seconds and +10 seconds).

References

1. Miele, A.; Wang, T.; and Melvin, W.W.: "Maximum Survival Capability of an Aircraft in a Severe Windshear," Rice University, Aero-Astronautics Report No. 213, 1986.

2. Miele, A.; Wang, T.; and Melvin, W.W.: "Gamma Guidance Schemes and Piloting Implications for Flight in a Windshear," Rice University, Aero-Astronautics Report No. 212, 1986.

3. Hinton, David A.:"Flight-Management Strategies for Escape From Microburst Encounters," NASA TM-4057, 1988.

4. Hinton, David A.: "Piloted-Simulation Evaluation of Recovery Guidance for Microburst Wind Shear Encounters," NASA TP-2886,DOT/FAA/DS-89/06, 1989.

5. Oseguera, Rosa M.: "A Simple, Analytical 3-Dimensional Downburst Model Based on Boundary Layer Stagnation Flow," NASA TM-100632, 1988.

6. Boeing Co.: "Wind Shear Training Aid. Volume 1 - Overview Pilot Guide, Training Program", Contract DFTAO-1-86-C-00005, Feb. 1987. (Available from NTIS as PB88 127 196.)

7. Miele, A.; Wang, T.; Tzeng, C. Y.; and Melvin, W. W.: "Optimal Abort Landing Trajectories in the Presence of Windshear," Rice University, Aero-Astronautics Report No. 215, 1987.

8. Proctor, F. H.: "The Terminal Area Simulation System, Volume I: Theoretical Formulation," NASA CR-4046, 1987.

9. Proctor, F. H.: "The Terminal Area Simulation System, Volume II: Verification Cases," NASA CR-4047, 1987.

Table 1 Batch Run Recovery Altitudes

Note: Numbers in parenthesis indicate the wind shear encounter altitude for a particular initial altitude and alert time.

INITIAL ALTITUDE (feet)	STRATEGY	ALERT TIME (Seconds)				
		-10	-5	0	+5	+10
		(59)	(111)	(159)	(203)	(250)
	MANUAL	21	39	18	22	153
300	FLIGHT PATH	21	40	81	120	239
	GLIDE SLOPE	21	53	84	134	239
		(161)	(212)	(261)	(306)	(354)
	MANUAL	12	13	20	88	312
400	FLIGHT PATH	12	31	85	198	344
	GLIDE SLOPE	12	31	75	187	344
		(264)	(315)	(363)	(411)	(460)
	MANUAL	3	11	45	226	450
500	FLIGHT PATH	3	30	148	329	404
	GLIDE SLOPE	3	30	146	286	450
		(366)	(416)	(466)	(515)	(566)
	MANUAL	41	57	154	388	555
600	FLIGHT PATH	41	108	271	403	474
	GLIDE SLOPE	41	108	240	468	487
		(466)	(516)	(568)	(619)	(672)
	MANUAL	149	160	288	553	655
700	FLIGHT PATH	149	241	395	469	580
	GLIDE SLOPE	149	241	339	483	483

Table 2 Average Recovery Altitudes in Piloted Runs

	Recovery Strategy					
	Manual		Flight Path Angle		Glide Slope	
Alert >	Reactive	Forward	Reactive	Forward	Reactive	Forward
Shear Scenario						
TASS	206	503	134	472	210	481
Analytical at 1640 ft.	62	309	39	307	62	303
Analytical at 3600 ft	53	409	53	401	81	397
All Shears	107	407	75	393	118	394
All Runs This Strategy	257		234		256	

All Reactive Alerts: 100 ft. All Forward Look Alerts: 398 ft.

Table 3 Analysis of Variance of Recovery Altitudes in Piloted Simulation

Source of Variation	Sum of Squares	Degrees of Freedom	Mean Square	Computed f	Critical F at Level of Significance	
					0.05	0.01
Shear	1538177.3	2	769088.7	343.6	3.0	4.7
Alert	8390651.9	1	8390651.9	3748.5	3.9	6.8
Strategy	39972.3	2	19986.2	8.9	3.0	4.7
Interactions:						
Shear/Alert	121489.4	2	60744.7	27.1	3.0	4.7
Shear/Strategy	29317.2	4	7329.3	3.3	2.4	3.4
Alert/Strategy	27832.8	2	13916.4	6.2	3.0	4.7
All Factors	9701.1	4	2425.3	1.1	2.4	3.4
Error	805824.0	360	2238.4			

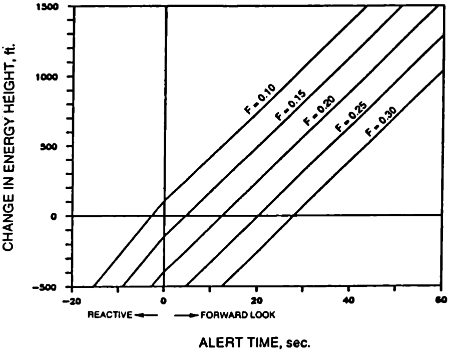


Fig. 1 Effect of Alert Time on Airplane Energy Height Change During a Wind Shear Encounter

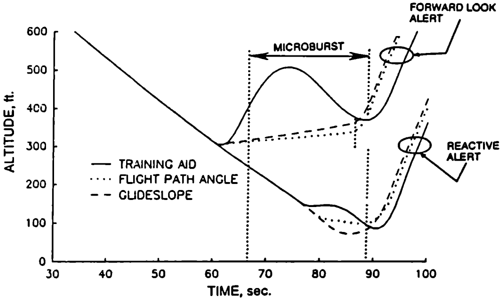


Fig. 2 Effect of Forward-Look Capability and Recovery Strategy Variation in Batch Simulation

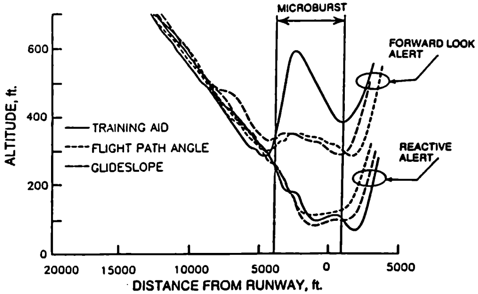


Fig. 3 Effect of Forward-Look Capability and Recovery Strategy Variation in Piloted Simulation

In-Flight and Ground Based Simulation:

Capabilities and Limitations

By

STEVEN R. MARKMAN

Flight Dynamics Laboratory

Wright-Patterson AFB OH

Abstract

Flight simulation has become an indispensable portion of the aircraft development process. Two types of simulation are in common use today: on the ground, using the familiar ground based simulator, or in actual flight, using a specially developed variable stability aircraft, commonly called an in-flight simulator. Ground based and in-flight simulation each have unique capabilities and advantages, along with their own limitations. For a specific task, one type may not be the best choice every time. This paper discusses major capabilities and limitations associated with ground based and in-flight simulation. It will also discuss the validity of simulation results.

Introduction

Aircraft and their associated systems have undergone a tremendous evolution since the end of World War II. They have progressed from propellers to jets, mechanical flight controls to fly-by-wire, and point designs to multi-role. Flying qualities used to be optimized around the primary mission requirement, with poor characteristics elsewhere accepted because there was no choice. Matching the machine to the pilot was unknown. It was common to hear that some aircraft could only be flown by "real pilots". The way we look at it today, the problem was in the design of the aircraft itself.

By the 1950s, with the introduction of jet engines, aircraft began flying higher, faster and farther. Power boost became necessary to operate the flight controls at high dynamic pressures and stability augmentation was developed to keep the control forces and aircraft dynamics acceptable to the pilot throughout the entire operating envelope. Avionics began to evolve from two way radio and raw navigation data into integrated communication, navigation and threat detection systems. The cockpit had to allow the pilot to operate in extreme environmental conditions, yet keep him comfortable for extended periods of time.

The modern aircraft of today and those on the drawing boards for tomorrow have flight envelopes and onboard systems that were probably beyond most peoples' imagination forty-five years ago. These resulted from breakthroughs in aerodynamics, propulsion, controls, materials and electronics, combined with the knowledge of how to use them. Modern aircraft represent massive compromises among the demands and restrictions of mission, performance, cost, and pilot ability. Requirements are often mutually exclusive, and the complete satisfaction of one may be at the expense of another. Often the pilot is the weak link, since the systems themselves are often better understood than how the human operator interfaces with them.

Analysis techniques tell us much about how an aircraft will perform. Unfortunately, they do not tell as much about how well the pilot can fly and maintain adequate control throughout the entire envelope for all required tasks. This is especially true of aircraft with higher order flight control systems. Manned flight simulation helps to resolve many of these problems, since the pilot (who is nonlinear and time varying) does not need to be modeled. By using manned flight simulation, aircraft and systems can be better designed around the capabilities of the pilot.

Just as aircraft have evolved, so too have the simulators used to develop them. Ground based and in-flight simulators have each developed their own areas of excellence, and they compliment each other very well when used in a combined test program. This paper concentrates on the different capabilities and limits of these two types of flight simulation.

Advantages/Limitations of Ground Based Simulation

Ground based flight simulation has become generally used today because it is a tool of almost limitless flexibility. Being on the ground, there are no safety of flight issues to be considered. New aerodynamic configurations and flight control designs can be simulated, and the new models easily updated as the design evolves. The operation of new systems can be evaluated even before breadboard circuits are designed, as can the effects of aerodynamic changes.

A typical ground based simulation usually consists of three pieces: a computer model of the aircraft and its systems, a cockpit, and a visual system (many also have a motion base). Other devices to provide cues to the pilot can be used to enhance the realism of the simulation. These include a g seat, g suit, vibrations, turbulence, and realistic sounds. Actual aircraft hardware can also be integrated into the simulation.

The simulation does not have to be complicated and highly detailed to be effective. A simplified simulation may answer a given question as well as a complex one. The first question that must be asked is: "What do I want the simulation to accomplish?" By answering this first, a simpler simulation may be found to be completely adequate. From a cost and schedule standpoint, it is desirable to mechanize the minimum amount. Extra detail may even cause anomalies which must then be accounted for. Unfortunately, there are no firm rules for determining how much detail is required. The experiences and judgment of simulation engineers is valuable in making these decisions.

A full fidelity simulation will usually incorporate a 6 degree-of-freedom aerodynamic model, a detailed cockpit, large amplitude motion, detailed visual scene, controls and displays for all systems of interest, sound, g suit, etc. The detail required of a given simulation will vary depending on it's purpose. For example, a flying qualities evaluation requires the aero model, feel characteristics, motion, and visual to be modeled accurately. Cockpit displays are of secondary importance, as long as the required information is displayed in a reasonable manner. In contrast, a simulation to investigate pilot interface with the cockpit would require detailed models of the systems operations, identical switches, and properly located displays. However, the aero model need not be exact as long as the task of flying the aircraft requires about the same workload. A sky/earth projector may provide an adequate visual scene if, say, the task is to operate the avionics equipment while maintaining level flight in turbulence.

Of course, the simulation performance can only be as good as the data provided to the model and the people putting the simulation together. For a new aircraft that is lacking actual flight test data for comparison, the quality of the simulation is very dependent on the experience of the people who derive the data and program the simulator.

It is essential to remember that simulation is not duplication. Although it requires the creation of as realistic an environment as possible, when real cues cannot be provided, then some type of substitutes must be given. This is especially the case with motion and visual cues.

The importance of motion cues has been debated over the years. The ground simulator obviously has a very limited displacement over which it can move. The motion base must produce an acceleration, then wash it out and return the cockpit to the central point without the pilot sensing the washout. The human sensory system is loaded with nonlinearities, making it extremely difficult to model. However, it can be fooled. To develop a technical base, many studies have been performed over the years to investigate motion washout algorithm design, looking at such things as motion detection thresholds and model acceleration to simulator initial acceleration ratio, to name only a few. The best that can be said is that motion algorithm design is not well understood, and greatly dependent on the skill, past experience, and intuition of the designer.

over the years to investigate motion washout algorithm design, looking at such things as motion detection thresholds and model acceleration to simulator initial acceleration ratio, to name only a few. The best that can be said is that motion algorithm design is not well understood, and greatly dependent on the skill, past experience, and intuition of the designer.

As the motion is washed out, the instruments, visual system, g seat and a g suit play an important part in maintaining the illusion of motion. Poorly designed motion may degrade pilot acceptability by introducing false cues in other axes. This may or may not give worse results than no motion. (The subject of no motion itself is interesting. Once the pilot gets used to the motion and stops thinking about it, he may not notice that it has been turned off. He may comment that the aircraft feels different, but without suspecting why.)

It is fairly well accepted that motion is not essential for many training tasks. However, in such a use, the objective is to maximize positive transfer of training. For engineering simulation, when minute changes in aircraft responses or piloting abilities are being sought out, the effects of motion can greatly influence pilot rating. This is especially true in handling qualities evaluations.

Motion systems are often "fine tuned" for a specific simulation based on pilot opinion. This must be done with great caution, or the pilot may interpret different motion cues as a change in aircraft dynamics. Motion that "feels good" may not really be the result of correct cues, but simply the ones that make the motion feel good to the pilot. A poor aircraft may feel better by changing motion cues, and vice versa. The correct cues are those that cause the pilot to perform at the same level with the same workload on a given task in a given aircraft as can be achieved in actual flight. For a simulation of a new aircraft that has not yet been flown, there is nothing for the pilot to compare to. As previously discussed, the judgment of the simulation engineers is critical.

There is little debate over the need for good visual cues, since they are essential for most piloting tasks. The problems with visual systems are mainly those of resolution, detail, field of view, and lack of depth.

Television projection equipment has a finite number of lines with which to make the picture, typically in the range of 500 to 1100. The picture is then projected around the cockpit on a screen. The human eye is capable of detecting very small objects, especially when they are moving. One problem with most visual display systems is that an object may not fill even one line width until at a relatively close range. Ground targets the size of a tank or truck may not even be seen until the attacking aircraft is closer than the normal pull up point. To get around this problem, systems have been developed with background and high resolution inset projection capability, or with multi-screen displays.

When a single, narrow field of view picture is projected ahead of the cockpit, pilots may complain about the lack of peripheral vision. They equate it to "flying through a tunnel" or "flying with blinders". Peripheral vision is important because it gives good cues for angular movements and especially forward velocity when flying at low altitudes. A field of view of 180 degrees will greatly improve peripheral cues.

Last, since a picture is projected on a screen, it is at a fixed viewing distance. The pilot must learn alternate techniques for judging depth. In doing so, he is learning to fly "simulator cues", rather than the real aircraft to which normally accustomed.

In-Flight Simulation

An in-flight simulator is an aircraft whose stability, feel, and flying characteristics can be changed to match those of another aircraft. This is accomplished by the pilot flying the aircraft through a "fly-by-wire" variable stability flight control system. A discussion of how the variable stability system operates is beyond the scope of this paper. However, the end result is that as the pilot moves the controls, the in-flight simulator aircraft responds as the simulated aircraft would. The pilot experiences the real flight motions and handling qualities of the simulated aircraft.

In-flight simulation has been used in the development of the F-16, YF-17, F-18, A-10, B-1, and Space Shuttle aircraft: X-15, X-24, and X-29 research aircraft: many types of generic research, especially handling qualities; and for several types of specialized training, particularly for test pilots.

The two best known in-flight simulator aircraft are the Flight Dynamics Laboratory's NT-33A (Fig 1) and NC-131H "Total In-Flight Simulator", or TIFS (Fig 2). Other in-flight simulators in the United States include NASA's Gulfstream II Shuttle Training Aircraft (Fig 3), the US Navy's X-22 (Fig 4), and the Air Force Test Pilot School/Navy Test Pilot School/Calspan Learjet (Fig 5). Others are operated throughout the world.

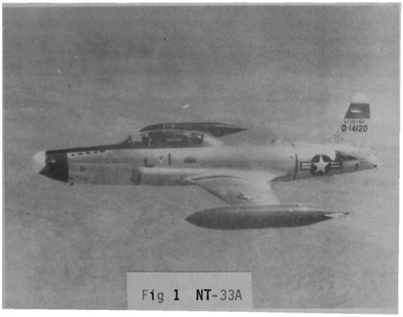


Fig 1 NT-33A

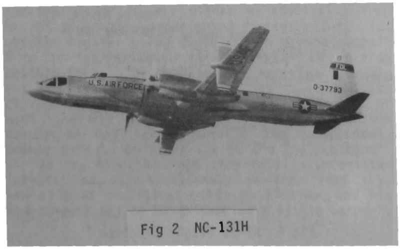


Fig 2 NC-131H

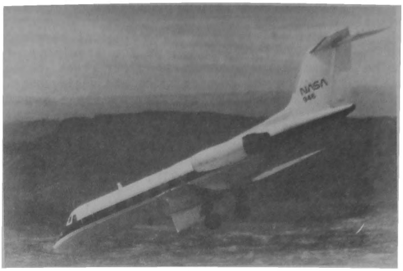


Fig 3 Shuttle Training Aircraft

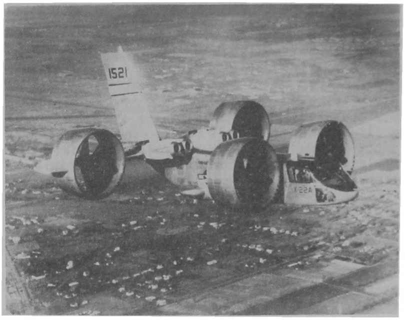


Fig 4 X-22

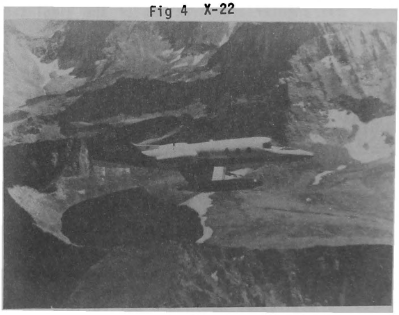


Fig 5 AFTPS/NTPS/Calspan Learjet

Advantages/Limitations of In-Flight Simulation

After considering the problems associated with the motion and visual systems of ground based simulators, it should be apparent that these two most important cues are readily available in an in-flight simulator. The motion is complete and sustained, and the visual is the real world. Indeed, in-flight simulation first came into use in the early 1950s because it was the only way to provide these cues realistically.

In addition to providing real motion and visual scene, the in-flight simulator overcomes what can best be called a "psychological barrier" in the pilot. This comes from the knowledge of being in an actual aircraft rather than being seated safely on the ground. Time and again, experience has shown that pilots tend to fly a ground simulator differently than they would an actual aircraft. Thus, poor handling characteristics, and especially pilot-induced oscillation (PIO) tendencies, have gone undetected. Specific examples of this are the YF-16 lateral PIO, YF-17 pitch PIO, and space shuttle pitch/roll PIO. YF-16 and space shuttle PIOs were observed in the actual aircraft. The potential for PIO in the YF-17 was discovered before first flight, changes were made to the flight control system, and the problem never appeared in the actual vehicle.

Although the reason for the greater realism is undoubtedly some combination of motion and visual cues and the real flight environment, the bottom line is that the in-flight simulator elevates the pilot gain to a more realistic level. This forces him into the mental realization that he is flying a real aircraft, thus demanding more precise and realistic control of the aircraft. With in-flight simulation, all of the concerns of actual flight are there and they are real: especially the knowledge that a sloppy landing could result in possible aircraft damage and personal injury.

Of course, just as certain factors affect ground based simulation, in-flight simulation may not be perfect either. A near perfect simulation should be achievable, in theory, if the in-flight simulator has complete control of all six degrees-of-freedom (DOF), a flight envelope equal to that of the simulated aircraft, and adequate control power and actuator dynamics to match all of the linear and angular accelerations of the simulated aircraft.

The evaluation pilot must feel the accelerations he would feel if seated in the cockpit of the simulated aircraft. Thus, independent control of all six degrees-of-freedom is essential to accurately reproduce the correct linear and angular accelerations, rates, and attitudes of the simulated aircraft's cockpit. This is a critical factor when simulating large flexible aircraft. Without 6 degrees-of-freedom, the farther the pilot is seated from the center of rotation the more noticeable any error between model and response will be. An in-flight simulator such as the TIFS, designed to simulate large aircraft, has to have 6 DOF control to match all required motions. In smaller in-flight simulators, 6 DOF control of course is desirable, but fewer may be tolerable, depending on the intended use. The NT-33A, which has only 3 DOF control, has been adequate for most fighter type aircraft simulations. This is because pilot location does not vary considerably in this size of aircraft, and the mismatch has been tolerable, although obviously undesirable. However, recent experience has shown that for multi-controller aircraft, the NT-33A's capabilities are very limited or inadequate altogether. For example, the NT-33A, lacking direct lift control, could not be used to simulate the X-29 research aircraft because of the blending of direct lift and pitch control incorporated in that aircraft.

Envelope matching is also highly desirable. If the in-flight simulator's performance envelope is less than that of the simulated aircraft, then only that limited portion where they overlap has the potential to be accurate. Although the NT-33A has simulated the F-16 and F-18 , its limited envelope (Fig 6) has restricted most of this work to the landing approach. Of course, it is not necessary to match actual airspeed and altitude in order to match handling characteristics and aircraft dynamics. In up and away flight, the instruments can be controlled independently to reflect the simulated flight condition. But this is not practical for low altitudes where ground cues will make the discrepancy apparent, or for air combat maneuvering with other aircraft, because the performance and maneuverability of the other aircraft may be highly dependent on flight condition. Speed mismatch can sometimes be compensated for in a landing simulation by inserting a headwind into the model.

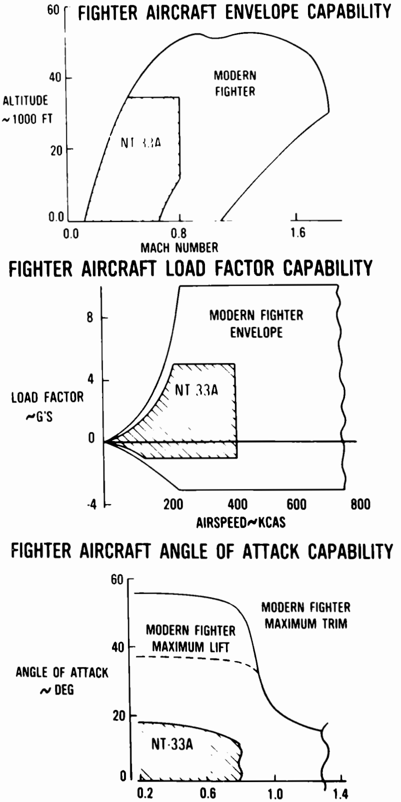


Fig 6 NT-33 Flight Envelope

Lack of control power may also cause a loss of simulation fidelity by making it impossible to match angular accelerations. Inadequate actuator dynamics may make it impossible to produce the required onset rates even if the steady state rates can be produced. Control power can be increased by modifying the hydraulics and actuators, or changing or adding additional control surfaces, all very expensive modifications to an aircraft.

Several other factors can affect in-flight simulation fidelity. These include equipment performance in an airborne environment, less flexibility in reconfiguring cockpit equipment, and weight, size, and power considerations in selecting airborne equipment. Last, the psychological barrier may not always be completely broken, since the evaluation pilot may be counting on the safety pilot to take over control if he gets into trouble.

Validity of Simulation Results

Until confirmed by flight testing the actual vehicle, simulation results, both ground based and in-flight, should be considered as predictions, at best. Although many studies have been performed over the years to compare simulation results to those of actual flight test, to date there is till no absolute correlation. There are many reasons to question simulation results, most of which apply to varying extent to both in-flight and ground based simulations. As already discussed, the physical limits of the simulators themselves are probably the biggest sources.

A good model made from good data is essential for proper aircraft and system response. Programming errors can ruin both, and mutually compensating errors are rare. Are sign conventions, definitions, and nomenclatures consistent (is aileron deflection measured relative to the wing chord line or from one aileron chord to the other)?

A good test plan is critical. It must cover all portions of the envelope being studied. The task required of the pilot must be representative of one that would be expected of him in the real aircraft. If either of these are inadequate, serious problems may go undetected.

Pilots should be chosen somewhat for the task at hand: qualifications and background should be considered when selecting subjects and interpreting results. For example, a highly experienced transport pilot's assessment of a new fighter's handling qualities may differ substantially from that of an equally experienced fighter pilot. But even the fighter pilot's evaluation may be suspect if he is inexperienced in the task. It is also possible that he has had so much experience in one particular aircraft that he has mastered its idiosyncrasies and is no longer really conscious of them. He may not give an objective comparison because he may not appreciate the improvement. Another type to be alert for is the "golden hand". This is the pilot who can step into an unfamiliar aircraft and fly excellently on the first try. Even a "bad" configuration is no problem, and his only comment may be he's "working a little harder".

Consider another hypothetical situation. The first flight configuration for a new aircraft has been frozen, but the control system design for future flights is still being studied. Which pilot is to evaluate the new configurations? The pilot selected to fly the first flight? If so, he may be training himself for the first flight, and not really interested in helping to make improvements for future flights.

Summary

Flight simulation, both ground based and in-flight, is the best accepted means for investigating the characteristics of new aircraft technologies and designs. Simulation is an integral part of the design process, but simulation results must be weighed in light of the capabilities of the simulator used. Many factors weigh the choice between ground based and in-flight simulators and the level of detail to be used. Careful thought must be given to which factors are critical by considering the desired end results of the simulation. In many cases, the use of both types of simulators in a combined flight simulation program is justified and desirable.

89-3329-CP

IN-FLIGHT SIMULATION OF UNSTABLE AIRCRAFT SYSTEMS

Masaki Komoda[*]
Tokyo Metropolitan Institute of Technology, Tokyo, Japan

Nagakatsu Kawahata[†]
Nihon University, Chiba, Japan

and

Yukichi Tsukano[¶] and Koki Hozumi[∂]
National Aerospace Laboratory, Tokyo, Japan

Abstract

Simulation errors associated with in-flight simulations of linear and unstable aircraft systems are discussed from flight mechanical viewpoint. It is pointed out that response feedback control schemes are desirable for simulating gust response of any models and control response of stable models but are not applicable for simulating control response of unstable models, and that explicit model-follow system must be exclusively used for simulating unstable models. Flight test results using a variable stability and response airplane are included to support the statement.

Introduction

In-flight simulator may be used for a variety of problems which are associated with the pilot-vehicle closed loop situation. Typical in-flight simulation of an aircraft system is shown in Fig.1. Evaluation pilots are supposed to control a prescribed mathematical model and actually control an in-flight simulator which is a hardware and is called a plant. If responses of plant to initial conditions, to control inputs and to external disturbances are the same as those of model, the simulation is satisfactory at least from flight mechanical standpoint. Should any mismatch exist between these two, they may be called simulation errors. If the simulation errors exceed an acceptable level, then fidelity in simulation will be lost. Interrelations between stability of given models and control law which is utilized for simulation are discussed in the followings.

Most of in-flight simulators are based upon model following theories (ref.1, 2, 3). Standard block diagram is shown in Fig.2. A class of control is the so-called 'response feedback (RFB)' and another is the 'explicit model following (EMF)'. Ref.3 showed that, under several conditions which are practically acceptable, an output error dynamics of model-following system can be arbitrarily assigned, and the characteristic roots of the assigned error dynamics usually coincide with those of plant's closed loop. Also, it is shown that a variety of combinations between feedforward gain K_{xm} and feedback gain K_x is possible while keeping the same model-follow error dynamics. The theory suggests that a model can be simulated by either feedback only (RFB) or any combination of feedback and feedforward (EMF). This is true when the model to be simulated is sufficiently stable, and the arbitrariness in choice of gains may be utilized for disturbance rejections for example, in addition to model following purpose (ref.4).

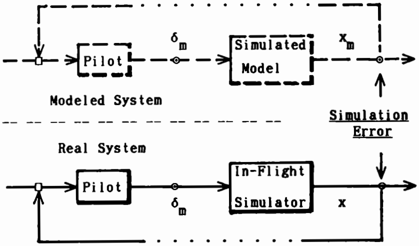


Fig.1 Simulation of aircraft model

By using RFB control scheme, model dynamics are simulated by plant's closed loop dynamics, which in turn characterize the model follow error dynamics. Therefore, when the model becomes unstable or marginally stable, so become the error dynamics and RFB system (implicit model-following system) fails. A flight test example using VSRA was reported in ref.5 where an unstable lateral-directional aircraft system was successfully simulated by EMF. Another flight tests for unstable longitudinal aircraft system and pertinent discussions are described below.

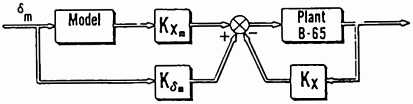


Fig.2 Model-follow control law

[*] Professor, Aerospace System Engineering, formerly National Aerospace Laboratory, Member AIAA

[†] Professor, Precision Machinery, formerly National Aerospace Laboratory, Member AIAA

[¶] Senior Researcher, Flight Research Division

[∂] Researcher, Flight Research Division

1. Simulation of Unstable Systems

Before analyzing simulation errors, let us consider typical in-flight simulations where unstable systems are simulated in connection with a failure which occurs in control systems.

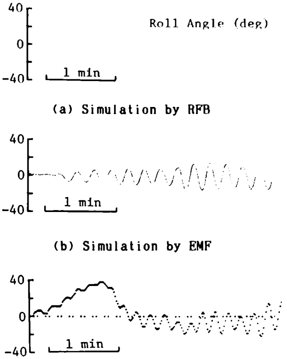
Marginally Controllable Systems

In-flight simulation is often planned to determine the controllability limit by pilots of an unstable aircraft system such as a relaxed static stability airplane with failed augmentation. The static margin K_n, may be considered a representative parameter for the test program. Whether or not pilots can manage to stabilize a model airplane of a given K_n, or how difficult the pilots' task is, would not be a simple function of K_n, but also depend on the initial conditions of the simulator. This is because evaluation pilots need some time for adapting himself to the change in the system characteristics, and during this period the unstable system starts to diverge depending upon its initial condition. Thus, pilots begin their positive control starting from some disturbed condition. From a stabilized initial condition, pilots may keep a stabilized closed loop as indicated in Fig.1. He may rate the configuration controllable and the workload as tolerable. On the contrary, since a pilot is not a linear element at all, if the simulator was considerably disturbed initially, he will not be able to bring the closed loop into a stable state. Thus he rates the same model uncontrollable and the required workload intolerable. If the degree of instability is mild, pilot's ratings would not be affected by initial conditions, whether starting from a good trim or considerable out-of-trim. The effect of this unstable transient to the ratings will be more pronounced when K_n approaches to its uncontrollable limit.

In the above, only the initial condition of the simulator was considered as an parameter that might affect pilots' rating. By 'simulation errors', let us mean any objectionable disturbances which are put into the system inadvertently and therefore may delude pilot's ratings. Simulator's response to atmospheric turbulence is one of simulation errors when the effect of gust is excluded from the simulation test purpose. When the simulation test is planned to include gust disturbance, then the simulator's gust response must not be taken as a simulation error, provided that the simulator's response matches the simulated airplane's gust response.

Failure Detection

In another application of our in-flight simulator, pilot's ability for detecting failures in control systems was studied (ref.5). At the first stage, pilots are asked to detect a failure where the inherent stability changes from normal (stable) to unstable while control effectiveness is kept normal. In the next stage, they are asked to detect control loss and also to isolate the failure mode. In the former case, the evaluation pilot concentrates on stabilizing the unstable system and at the same time on identifying whether the required workload has increased or not, and thus tries to detect the failure. During this phase of detection, any simulation errors lead to a possible misdetection. In the latter case, free responses with zero control inputs must be simulated without any contamination of simulation errors.



(a) Simulation by RFB

(b) Simulation by EMF

(c) Actual flight data (ref.6)

Fig.3 Unstable lateral-directional mode

Fig.3 shows typical flight test results using VSRA to simulate an aircraft accident where the dutch roll mode became slightly unstable and lateral-directional controls were completely lost (ref.5). Compared are simulations using (a) RFB and (b) EMF. In Fig.3(c), actually recorded flight data from DFDR in the accident (ref.6) are compared. What was supposed in the simulation tests is an immediate and gradual onset of dutch roll oscillation as is seen in Fig.3(c). Firstly the RFB was used but resulted in a poor simulation in that the dutch roll mode was not clearly seen. Then the EMF was tried and gave relatively clear onset of free oscillation.

2. Simulation Errors

General Discussions

In view of the stability and control response characteristics, simulation fidelity is evaluated by the errors between model and plant outputs. Given a linearized equation of motion of a simulated airplane (model),

$$\dot{x}_m = A_m x_m + B_m \delta_m + G \dot{\xi}_m ; \quad x_{m0} \tag{1}$$

and of a simulating airplane (plant),

$$\dot{x} = A x + B \delta + G \dot{\xi} + (\dot{x})_0 ; \quad x_0 \tag{2}$$

respectively (ref.5), where x, x_m are state variables, δ, δ_m control variables, and ξ, ξ_m are gust components under the point approximation (See foot note). When one refers a model, the initial condition $x_{m0} = x_m(0)$ as well as eq.(1) itself is implied. When the plant dynamics are observed by perturbations of state and control variables from their referenced values at the instant of system engagement, an out-of-trimmed state provides an apparent step input to airplane dynamics. A term $(\dot{x})_0 = A\,x_0 + B\,\delta_0$ is included in eq.(2) to indicate this effect in actual flight.

Just for simplicity, two systems A and A_m, and B and B_m are assumed to have the same dimensions (n) ×(n) and (n)×(m), respectively. Assume a control law

$$\delta = -K_x\,x + \delta_f, \quad \delta_f = K_{xm}\,x_m + K_{\delta m}\,\delta_m \tag{3}$$

to construct a model following system to simulate the model, where a feedback gain K_x, a feedforward gain K_{xm}, and a direct gain $K_{\delta m}$ are appropriately chosen. (See Fig.2.)

In order to evaluate fidelity of simulations, let the simulation error be defined by

$$\varepsilon = x - x_m\,. \tag{4}$$

Using eq.s (3) and (4), the elimination of x from eq.s(1) and (2) yields,

$$\begin{bmatrix}\dot{\varepsilon}\\ \dot{x}_m\end{bmatrix} = \begin{bmatrix}A_c & \Delta F\\ 0 & A_m\end{bmatrix}\begin{bmatrix}\varepsilon\\ x_m\end{bmatrix} + \begin{bmatrix}\Delta H\\ B_m\end{bmatrix}\delta_m + \begin{bmatrix}G(\dot{\xi} - \dot{\xi}_m)\\ G\dot{\xi}\end{bmatrix}$$
$$+ \begin{bmatrix}(\dot{x})_0\\ 0\end{bmatrix} ; \quad \begin{bmatrix}\varepsilon_0\\ x_{m0}\end{bmatrix} = \begin{bmatrix}x_0 - x_{m0}\\ x_m(0)\end{bmatrix} \tag{5a}$$

where abbreviations are defined as follows.

$$\begin{aligned} A_c &= A - B\,K_x\\ \Delta F &= (A - B\,K_x) - (A_m - B\,K_{xm})\\ \Delta H &= B\,K_{\delta m} - B_m \end{aligned} \tag{6}$$

If x_m, instead of x, is eliminated from eq.s (1), (2), the first equation of (5a) becomes

$$\dot{\varepsilon} = (A_m - B\,K_{xm})\,\varepsilon + (\Delta F\,x + \Delta H\,\delta_m) + G(\dot{\xi} - \dot{\xi}_m)$$
$$+ (\dot{x})_0\ ; \quad \varepsilon_0 = x_0 - x_{m0} \tag{5b}$$

which is exactly equivalent to eq.(5a). If each gains K_x, K_{xm} and $K_{\delta m}$ are designed based upon a plant model (A',B'), which is usually more or less deviated from the real plant (A,B), then A'_c, ΔF', and ΔH' must be substituted in place of eq.(6),i.e.

$$\begin{aligned} A_c' &= A_c + (\Delta A - \Delta B\,K_x) = A_c + \Delta A_c\\ \Delta F' &= \Delta F + [\Delta A - \Delta B\,(K_x - K_{xm})] = \Delta F + \Delta_a F\\ \Delta H' &= \Delta H + \Delta B\,K_{\delta m} = \Delta H + \Delta_a H \end{aligned} \tag{6'}$$

where the modeling error of plant is defined by

$$\Delta A = A' - A\ , \quad \Delta B = B' - B\ . \tag{7}$$

Eq.(5a), or equivalently eq.(5b), shows the total modes of simulation error. It will be convenient to subdivide ε(t) into three parts, i.e. (i) free response ε_f to the initial condition ε_0, (ii) inherent model follow error ε_δ due to ΔF and ΔH, and additional model follow error due to plant modeling errors $\Delta_a F$ and $\Delta_a H$, (iii) erroneous gust response ε_g due to $\xi - \xi_m$, and (iv) out-of-trim effect ε_t due to $(\dot{x})_0$.

Let us assume temporarily that $\xi = \xi_m = 0$ and $(\dot{x})_0 = 0$. Those effects will be considered later. The solution of eq.(5a) is obtained after Laplace transform,

$$E(s) = (sI - A_c')^{-1}[\Delta F'(sI - A_m)^{-1}B_m + \Delta H']\,D_m(s)$$
$$+ (sI - A_c')^{-1}[\Delta F'(sI - A_m)^{-1}x_{m0} + \varepsilon_0]. \tag{8}$$

From eq.(8), one can see that; (a) a part of ε_δ and ε_f due to $\Delta H' D_m(s)$ and to ε_0 evolves with inherent modes of A_c only, and (b) another part of ε_δ and ε_f due to ΔF' and x_{m0} evolves with inherent modes of both A_c and A_m.

Should exact conditions

$$\Delta A = \Delta B = \Delta F = \Delta H = 0 \tag{9a}$$

hold, then simulation error dynamics in eq.(5a) or (5b) is decoupled from the model or plant dynamics, and ε(t) solely depends on its initial condition;

$$\begin{aligned} E(s) &= E_f(s)\\ &= (sI - A_c)^{-1}\varepsilon_0; \quad \varepsilon_0 = x_0 - x_{m0}. \end{aligned} \tag{10}$$

In practical applications, eq.(9a) is too optimistic and hardly satisfied. In order that the simulation error be almost free from the control input δ_m, it will be easy to see from eq.s (5a,b) or (8) that, the conditions

$$\Delta F' \sim 0, \quad \text{and } \Delta H' \sim 0 \tag{9b}$$

and

$$A_c' : \text{sufficiently stable,} \tag{9c}$$

(note) When aircraft equations of motion are written with reference to airmass, and the frozen gust gradients along aircraft size are neglected, the effect of gust is represented by the time derivative of gust components u_g and w_g, which are designated as $\dot{\xi}$.

are mandatory, where the symbol ~ is used to indicate 'nearly equal to'. The requirement of eq.(9b) is most conventionally satisfied by using the minimum norm criteria (ref.1), i.e.,

$$K_x = (B^TB)^{-1}B^T[A - (A_m - B K_{xm})]$$
$$K_{\delta m} = (B^TB)^{-1}B^TB_m . \qquad (11)$$

If the number (m) of available control is insufficient, B does not span the ranges of $(A - A_m)$ and of B_m and therefore, ΔF and ΔH remain nonzero. Modeling errors ΔA and ΔB aggravate these effect, and thus produce $\varepsilon_\delta(t)$. The model following error ε_δ is the one which must be avoided to obtain fidelity of simulation because it couples with δ_m, and leads to different control response characteristics. The model following errors will be discussed below.

Explicit Model Following System

In an explicit model following system, model computation as per eq.(1) is conducted on-line. The gains K_x, K_{xm} and $K_{\delta m}$ are so determined that the conditions

$$A_c \sim (A_m - B K_{xm}), \quad B K_{\delta m} \sim B_m \qquad (12)$$

are satisfied. Since $B K_{xm}$ modifies A_m to some extent, no direct connection is called for between A_m and A_c, and one can assign a sufficiently stable closed loop A_c independently of whether A_m is stable or unstable. Examining eq.(5a) or (8), it will be seen that; (i) the free response ε_f to ε_0 simply decays, and therefore a simulation for arbitrary initial condition x_{m0} is possible, (ii) the forced response ε_δ to $\delta_m(t)$ through ΔH' can be suppressed within an acceptable boundary by a stable A_c, (iii) the forced response ε_δ to model state $x_m(t)$ through ΔF' is again expected to be suppressed by a sufficient stability attached to A_c even if A_m is unstable.

Response Feedback

When the feedforward loop is deleted from eq.(3)

$$K_{xm} = 0 , \qquad (13a)$$

on-line model computation is not necessarily needed. Eq.(3) gives a RFB control law. Eq.(12) becomes now

$$A_c \sim A_m , \quad BK_{\delta m} \sim B_m . \qquad (13b)$$

Free response ε_f of simulation error is considered first. It is given by the second term of RH of eq.(8), or more directly from eq.s (1) and (2),

$$E_f(s) = (sI - A_c)^{-1}x_0 - (sI - A_m)^{-1}x_{m0} . \qquad (14)$$

Needless to say, $\varepsilon_f(t)$ dies out quickly when the model A_m, and hence the closed loop of plant A_c, is sufficiently stable. When A_m and A_c are unstable or marginally stable, $\varepsilon_f(t)$ may affect simulation results because of the reasons stated before.

In a 'response feedback' system, the on-line model computation has been discarded from the beginning. So, when considering eq.(8) or (14), a question arises; what does x_{m0} in ε_0 stand for? It will be adequate to note here that the initial condition of plant x_0 varies from a test case to another case, because $x_0 \neq 0$ is easily brought about due to out-of-trim condition at the instant of system engage. When the system is engaged by a safety pilot, each realization of x_0 is a random and unknown variable from the evaluation pilot's standpoint. The only way to estimate unknown quantities x_0 and $(\dot{x})_0$ in eq.(2) is post flight data reconstructions using least square method.

Two interpretations for x_{m0} may be possible. The one is to accept that $x_{m0} = x_0$ has been assumed for the simulation task whatever x_0 may be. Upon this interpretation, eq.(14) implies that diverging $\varepsilon_f(t)$ comprises the difference between eigenvectors of A_c and A_m if any. In fact, eigenvalues of A_c can usually be set almost equal to those of A_m, but eigenvectors of A_c do not match those of A_m especially when only small number (m) of controls are available. Free responses depend upon each initial conditions. For example, the spiral mode dominates in Fig.3(a), whereas in Fig.3(b) dutch roll mode is clear, but it must be noted that they include somewhat erroneous $\varepsilon_f(t)$. The other interpretation is to assume a particular value of x_{m0}(e.g. $x_{m0} = 0$), and hence to accept the fact that each simulation test is contaminated by the first term of RH of eq.(14). Many test data will have to be screened out because of unacceptable initial conditions $x_0 \neq 0$.

Forced responses of simulation error will be considered next. The eigenvalues p'_k of A'_c are anyhow close to those q_k of A_m, i.e., $p'_k \sim q_k$; k=1,..,n. From the discussions next to eq.(8) above, a part of $\varepsilon_\delta(t)$ responds like $\exp(p'_k t)\cdot\Delta h'_{ij}$ but another part of $\varepsilon_\delta(t)$ which is proportional to ΔF' shows double pole characteristics, or in other words, they evolve like $\Delta f'_{ij} t\cdot\exp(p'_k t) * \delta_{ml}(t)$, where $\Delta f'_{ij}$ and $\Delta h'_{ij}$ are entries of ΔF' and ΔH', respectively, and ( )*( ) indicates a convolution. Again, if a stable model A_m is simulated, then A_c is stable and the simulation error $\varepsilon_\delta(t)$ decays as time elapses. Therefore, one can expect a good simulation test. Contrary, if an unstable model A_m is simulated where $\mathrm{Re}(p'_k)$ is made positive to match

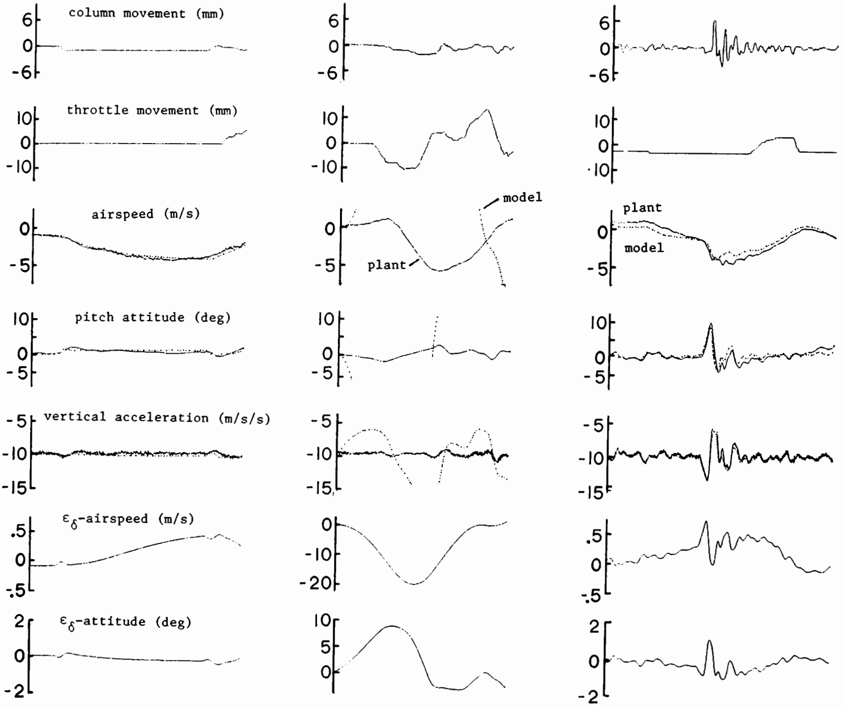
Re(q_k) for some k's, the simulation error $\varepsilon(t)$ never decays but diverges with modes exp($p'_k t$) and t·exp($p'_k t$). The situation is worse than that with a free response only, because the rate of divergence is expedited by the unstable double poles. Above all, large simulation errors dependent on the input $\delta_m(t)$ are included in the plant states which evaluation pilots are observing to producing their controls. The fact gives rise to poor fidelity in simulation.

Gust Response

In-fligh simulators are exposed to real atmospheric turbulence ξ, and it is of advantage when the turbulence effect is included in the simulation. In such a case, one can assume that $\xi_m = \xi$, and $\varepsilon_g = 0$. For that, in EMF, on-line measurement $\hat{\xi}$ of $\xi (= \xi_m)$ is required to be fed into model computation in eq.(1). The magnitude of simulation error ε_g is dependent on the precision of turbulence measurement ($\hat{\xi} - \xi$), and the stability of A_c. On-line measurement of turbulence needs sophisticated implementation and data handling. Therefore, RFB scheme is preferable from gust response standpoint, where the turbulence measurements are not required. The condition $\xi = \xi_m$ holds exactly, and the simulation error ε_g is excited through $\Delta F'$ and its magnitude depends on whether A_m is stable or unstable.

3. Flight Test Examples

Flight tests are conducted using Variable Stability and Response Airplane (VSRA). The test program was intended to determine the controllability limit of statically unstable aircraft by both attitude command and attitude rate command in longitudinal axis. The unstable model characteristics are adapted to get K_n= -3 % of the model in Table A-1 of ref.8, and are summarized in Table 1.



(a) Stable system by RFB (b) Unstable system by RFB (c) Unstable system by EMF

Fig.4 Comparison of flight test results for unstable longitudinal system

In expectation of more realistic gust response, RFB type model follow system was favorably chosen. Having found that larger simulation errors result, explicit model following control was utilized. Combination of a stable model and EMF has been presented elsewhere (ref.5), and is not included.

In the following, only the effects of model's stability and of the control scheme to the resulting simulation errors will be elucidated. For comparison purpose, simulation of a stable system by RFB is included. The stable model corresponds to an augmented model of the unstable model. Its characteristics are also summarized in Table 1.

Table 1. Model parameters (ref.8)

	Stable model	Unstable model
$1/T_{sp1}$	-.65	-.85
$1/T_{sp2}$	-.067	.094
ω_p	1.4	.19
ζ_p	.47	.59

Details of VSRA are described in ref.5. Although VSRA is provided with DLC capability, only elevator and thrust controls are used for simulations. The VSRA's control law is based on an explicit model follow theory (ref.3) and summarized in ref.5.

Evaluation pilots are requested to maintain steady and level flight, to reduce flight speed by 10 mph and then to recover while keeping level flight. In Fig.4, compared are typical flight records of, (a) a stable system by RFB, (b) an unstable system by RFB, and (c) an unstable system by EMF. Each columns include, pilot's input to the manipulators $\delta_m(t)$, plant state variables $x(t)$ as compared with model state variables $x_m(t)$, and inherent and additional simulation errors $\varepsilon_\delta(t)$ due to $\Delta F'$ and $\Delta H'$. Derivatives Z_q, $Z_{\delta e}$ and equivalent lag in thrust build-up are included as modeling error ΔA and ΔB in computing $\varepsilon_\delta(t)$ by eq.(5b). The model states in simulations of stable model (Fig-.4(a)) or by EMF (Fig.4(c)) are those computed onboard. It will be seen that the total simulation errors which are defined by the differences between $x(t)$ and $x_m(t)$ are small and correspond to $\varepsilon_\delta(t)$. Contrary, for simulation of the unstable system by RFB (Fig.4(b)), model states computed on-board simply diverge, but such model states have no practical meaning for RFB. Least square estimation of x_{m0} and $(\dot{x}_m)_0$ is mandatory to get best match of $x_m(t)$ with $x(t)$ in Fig.4(b). The total simulation errors are still so large that $x_m(t)$ is not fully shown in the ordniate scale. In the figure, ordinates of $\varepsilon_\delta(t)$ is 20 times for air speed and 5 times for attitude as compared with Fig.4 (a) and (c). Certainly, another causes such as gust response $\varepsilon_\xi(t)$ may be of importance too. However, without getting into more details along this line, it should be understood that simulating unstable systems by RFB control law is very likely contaminated by simulation errors due to various origins. Actually, evaluation pilots unavoidably control a plant of somewhat different transfer functions from originally supposed. Fig.5 depicts this situation as compared with Fig.1.

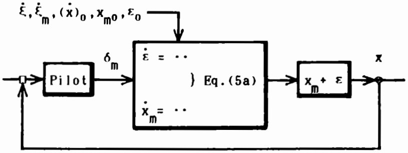

$\dot{\xi}, \dot{\xi}_m, (\dot{x})_0, x_{m0}, \varepsilon_0$
Pilot
δ_m
$\dot{\varepsilon} = \cdot\cdot$
} Eq.(5a)
$\dot{x}_m = \cdot\cdot$
$x_m + \varepsilon$
x

Fig.5 Simulated unstable model by RFB

As is seen from eq.(5b), the model-follow error grows up when the plant state x(t) is large. This model-follow error may considerably affect pilot's ratings in marginal cases such as investigation of controllabilty limits, wherein lack of stability in man-vehicle closed loop leads to large x(t). Under such situations, what is much worse is that the validity assessment of tests by post-flight data reconstructions is not an easy matter.

Concluding Remarks

Simulation errors associated with in-flight evaluation are generally discussed. It is pointed out that if a response feedback control scheme is used for simulating unstable aircraft systems, both free response to initial conditions of simulator and model-follow error to control inputs diverge, so that pilot ratings to the simu- lated model may certainly be affected. This is because the errors between plant and model are governed by characteristic roots of plant closed loop which are unstable. Explicit model-follow control scheme must be used to avoid such situations. Flight test data using VSRA in-flight simulator were presented to show the above statements.

References

*1 Erzberger,H.,"Analysis and Design of Model-Following Control Systems by State Space Techniques,"Proceedings of Joint Automatic Control Conference, University of Michigan, Ann Arbor,June 1968,pp.572-581

*2 Motyka,P.R., Rynaski,E.G. & Reynolds,P.A., "Theory and Flight Verification of the TIFS Model-Following System,"AIAA Journal of Aircraft,Vol.9, No5, May 1972

*3 Kawahata,N.,"Model-Following System with Assignable Error Dynamics and Its Application to Aircraft,"Journal of Guidance and Control,Vol.3, No.6,1980,pp.508-516

*4 Kawahata,N.,"Reply by Author to J.R.Broussard and L.E.Mabius," AIAA Journal of Guidance and Control, Vol.5, No.2,March-April 1982,p.219

*5 Komoda,M., Kawahata,N., Tsukano,Y. and Ono,T. "VSRA In-Flight Simulator--Its Evaluation and Applications," AIAA-88-4605-CP, Sept.7-9,1988 /Atlanta,Georgia

*6 Aircraft Accident Investigation Report---JAL 747SR, JA8119, Report No. 62-2(in Japanese)

*7 Etkin,B.,Dynamics of Atmospheric Flight, John Wiley & Sons,Inc., New York, 1972 pp.544-547

*8 McRuer,D. and Myers,T.T.,"Flying Qualities of Relaxed Static Stability Aircraft -- Vol.II, "DOT/FAA/CT-82 /130-II

89-3330-CP

VISTA/F-16 Design Features

Capt M. V. Dunbar[*] and Lt J. L. Dargan[**]
Wright Research Development Center (AFSC)
Wright-Patterson Air Force Base, Ohio

Abstract

The Variable Stability In-Flight Simulator Test Aircraft (VISTA)/F-16 is the next generation U.S. in-flight simulator for high-performance aircraft. The host aircraft for VISTA should have sufficient performance to simulate a wide range of aircraft dynamics. Design of an in-flight simulator requires integrating a variable stability system (VSS) with the existing host aircraft flight control system. For VISTA this involves interfacing new VSS computers with the existing F-16 digital fly-by-wire control system. This also requires a new safety system to ensure that the VSS/host aircraft system is as reliable as the host aircraft.

Nomenclature

A/D	Analog to Digital Converter
ATIS	Airborne Test Instrumentation System
DOF	Degrees of Freedom
DEPC	Direct Evaluation Pilot Control
DFCS	Digital Flight Control System (Host F-16 FCS)
E.P.	Evaluation Pilot
FCS	Flight Control System
HUD	Heads-Up Display
I/O	Input, Output
ISA	Integrated Servo-Actuator
MFD	Multi-Function Display
NIC	Network Interface Controller
OFP	Operational Flight Program
PROM	Programmable Read Only Memory
RFB	Response Feedback
RT	Remote Terminal
S.P.	Safety Pilot
TIFS	Total In-Flight Simulator
VIM	VISTA Integrity Management
VISTA	Variable-stability In-flight Simulator Test Aircraft
VSS	Variable Stability System

Background

The Variable Stability In-Flight Simulator Test Aircraft (VISTA) is an F-16D that will be modified to act as a high performance in-flight simulator. VISTA will replace the Air Force's current high performance in-flight simulator, the NT-33A, which is becoming logistically unsupportable and is no longer representative of future high performance vehicles.[1]

The key to VISTA's ability to simulate other aircraft is the variable stability system (VSS), which will compute control commands to the F-16 actuators to simulate the flight characteristics of the vehicle under evaluation. The VSS will use the response-feedback (RFB) technique for matching the motions of the aircraft being simulated. Gain scheduling will be incorporated to account for nonlinearities in the aerodynamics.

VISTA has four primary applications. The first is in-flight simulation for new aircraft development and pre-first flight flying qualities test. The second is training test pilots in the Air Force and Navy test pilot schools. The third is flight control research on flying qualities, flight control systems and display technologies. The fourth is flight vehicle integration studies of combat maneuvering and avionics/flight control system integration. Figure 1 shows examples of possible VISTA applications.[2]

The VISTA program contract was awarded to the contractor team of General Dynamics, Ft Worth Division, and Calspan Corp in July 1988. General Dynamics is the prime integrating contractor. VISTA is scheduled to enter flight testing in late 1990 and become operational in the fall of 1991.

The main engineering effort of the VISTA program is to develop the VSS and integrate it with the F-16 Digital Flight Control System (DFCS). The major tasks that this paper will address are the development of the computer interfaces and the safety system.

[*] Flight Controls Engineer, AIAA Member

[**] Electrical Engineer

Performance Enhancements

Use of an F-16D as the host aircraft for the next generation in-flight simulator will provide simulation performance improvements over the existing NT-33A.

Flight Dynamics Simulation Capability

The most important capability is the ability to simulate a wide range of aircraft dynamics. In terms of handling qualities parameters, the VISTA minimum capabilities are:

Short Period Natural Frequency	0 to 15 rad/sec
Short Period Damping	-0.1 to 1.1
N_Z/α	1 to 100 g's/rad
Stick Force/g	1 to 200 lbs/g
Phugoid Natural Frequency	0 to .5 rad/sec
Phugoid Damping	-0.3 to 1.0
Dutch Roll Natural Frequency	2 to 8 rad/sec
Dutch Roll Damping	-.1 to 1.0
Roll/Spiral Mode Natural Frequency	0 to 5 rad/sec
Roll/Spiral Mode Damping	-.1 to 1.0
Roll/Sideslip Φ / β	0 to 10
Variable Time Delay	.01 to .5 sec
Lead/Lag	1 to 63 rad/sec

The dynamic operating envelope is:

Sideslip	±10°
Side Acceleration	± .5g's
Landing Speed	130-160 Kts
Direct Lift	1 g
Pitch Pointing	$\pm 7^\circ$ $\Delta\alpha^{0}$
Normal Accel	-2.4/7.33 g's

These limits are applicable only in parts of the flight envelope. At lower speeds the faster dynamics can not be achieved due to a lack of dynamic pressure. The faster dynamics can not be achieved at higher speeds due to aeroelastic deflections and structural limits. The dynamic operating envelope is compared with existing U.S. in-flight simulators (the Total In-Flight Simulator (TIFS), the NT-33, and the Calspan Learjet) in Figure 2. The positive normal acceleration and roll rate capability of each aircraft is compared, and the maximum g, thrust to weight and roll rate capability of the X-29 is included for comparison. The thrust to weight ratios are shown for maximum thrust without afterburner at 10,000 ft. 'Direct Lift' is the capability to increase or decrease the normal acceleration without pitching the aircraft. It is a measure of the simulator's capability to command heave motions or to simulate different lift-curve slopes.

VISTA Simulation Envelope

The baseline VISTA simulation envelope is defined to be 80-90% of the F-16D envelope and less than Mach 0.9. This represents a significant increase over existing U.S. in-flight simulators. Figure 3 shows the flight envelopes of the in-flight simulators.

VISTA Maneuver Capabilities

The basic F-16 can achieve very fast pitch and roll rates. The mechanical flow restrictors on the Integrated Servo Actuators (ISA's) will be replaced by command limiters in the flight control computer. Therefore the deflection rates of the control surfaces can be increased to quicken aircraft angular acceleration response.

The planned aircraft angular accelerations are:

Roll Acceleration	6.0 rad/s^2 in 0.8 s
Pitch Acceleration	1.0 rad/s^2 in 0.4 s
Yaw Acceleration	0.44 rad/s^2 in 0.27 s

System Architecture

System Overview

As stated earlier, the VSS is the key to VISTA's ability to simulate other aircraft. The VSS will compute control surface commands based upon the control laws of the aircraft being simulated and will provide artificial feel characteristics through the variable feel centerstick in the evaluation cockpit. These VSS computed artificial feel characteristics and simulation control laws provide the evaluation pilot with the dynamics of the simulated aircraft. The basic VISTA architecture is shown in Figure 4.

The VSS uses three Rolm Hawk 32 computers and one Titan microcomputer. Functionally, the VSS Hawk computers contain software for response-feedback flight control calculations, VSS safety monitoring, and display and simulation management. The Titan computer contains the software for controlling the artificial feel characteristics for the evaluation cockpit centerstick. All VSS software is written in Ada, and all VSS computers have software to perform Built In Testing.

The VSS is interfaced to the F-16 avionics, display, and Digital Flight Control Systems (DFCS). It is the VSS interface with the F-16 DFCS which allows the VSS simulation control surface commands to be routed through the DFCS to the control surfaces. Thus, with this design, the F-16 DFCS actually outputs the VSS control surface commands to the control surfaces. To accommodate this VSS/DFCS interface, software and minor hardware changes will be made to the production F-16 DFCS.

VSS Computers

Hardware Specifications

Each of the Hawk computers has a speed of 1.4 million whetstones/sec, 8 megabytes of random access memory (RAM) expandable to 32 megabytes, and a cache memory of 128,000 bytes. The Titan computer uses an Intel 80386 CPU. All the computers are mil-spec, and qualified to 9 g's.

Software Functions

There are four categories of VSS software. The categories are 1) redundant VSS flight control software, 2) non-redundant VSS flight control software, 3) VSS safety trip/monitor software, and 4) general purpose VSS software. Redundant VSS flight control software performs functions most critical to control of the VISTA in the simulation mode and requires dual-redundancy for failure detection. An example is the response feedback software, which will be contained in both Hawks #2 and #3. Non-redundant VSS flight control software performs functions which are not critical to safe flight, and, therefore, do not require redundancy for failure checking. For example the software for implementing the model flight control system will be contained in Hawk #1 only. VSS safety trip/monitor software will monitor aeroservoelastic oscillations, compare sensor data, and monitor the centerstick for a hardover failure. The software for monitoring ASE oscillations and for monitoring the centerstick will be resident in Hawk #1. The software for comparing sensor data will be resident in Hawk #2 and Hawk #3. General purpose VSS software performs all other functions not critical to flight control or safety trip/monitoring. This software will reside in Hawk #1 and includes the display software, simulation management software, and ground simulation software[2].

The VSS Titan computer contains software for controlling the force/motion dynamics of the centerstick during simulation. During initialization, the Titan receives desired stick dynamics parameters from the Hawk computers. From these parametersand simulation pilot applied forces, the Titan computes the centerstick motion and sends the centerstick force and position data back to the Hawks.

BIT Testing

The VSS Hawk computers contain built-in-test (BIT) software to ensure that the computers are functioning properly. The Hawk BIT runs automatically when power is applied to the Hawk or when the Hawk is reset. There are three levels of BIT testing that can be accomplished in the Hawk. Level 0 takes less than ten seconds to execute and is a basic go/no go test which may not detect many of the possible failures. Level 1 takes several minutes to execute and can isolate a problem to the Field Replaceable Unit (FRU) level. Level 2 is a thorough test of Hawk computer logic that requires up to one hour to complete. The BIT testing for the Titan runs automatically when power is applied and performs a functionalitycheck on all the Titan circuit cards.

Software Tools

The VSS software will be developed with an off-the-shelf Ada programming support environment. This software environment will support other higher order languages such as Fortran and Pascal to accommodate future user-provided software. The capability to interface the different languages within a software program will be provided. Ada will also be used as the program design language.

Interface Design

The VISTA design required the addition of several new Mux busses to the production F-16 Mux architecture to support the VSS interface with the F-16 avionics, display, and digital flight control systems. As such, the DFCS was moved to the B-Mux to unload the A-Mux and the D-Mux; the F-Mux was added to provide a direct VSS to DFCS interface; the V-Mux and R-Mux were added to support simulation and recording. All electronic warfare equipment was removed from the B-Mux. Figure 5a shows the production F-16 Mux architecture, while Figure 5b shows the VISTA changes to the architecture.

The overall VISTA Mux bus architecture is shown in Figure 6. As depicted in this figure, VSS Hawk #3 is interfaced to the F-16 DFCS via a new, dedicated 1553B Mux bus called the F-Mux. This interface will allow for VSS engagement and disengagement as well as provide safety monitoring functions while in the simulation mode so that host F-16 operational flight limits are not exceeded. By dedicating the F-Mux for communication solely between VSS Hawk #3 and the DFLCC, the transport delay of the VSS commands to the control surfaces is minimized, thereby reducing the effect that transport delay has on simulation fidelity.

VSS Hawk#2 is interfaced to the F-16 1553B avionics Mux bus, the A-Mux, so that the VSS can be provided with F-16 sensor information.

Hawk #1 interfaces to the 1553B display Mux bus, called the D-Mux, permitting VSS simulation information to be displayed on the F-16 multifunction displays and HUD. The I/O chassis interface to the 1553B recorder Mux bus, called the R-Mux, permits VSS software parameters to be recorded on the AR700 data recorder. The I/O chassis interface to the 1553B VSS Mux bus, called the V-Mux, provides artificial feel characteristics to the evaluation cockpit centerstick. The I/O chassis is essentially an expansion of Hawk #1. It allows more bus controller cards , A/D cards and D/A cards to be connected to Hawk #1 via a dedicated Rolm Hawk interface bus.

Impact on DFCS Software

The VSS/DFCS interface requires software modifica- tions to the production F-16 DFCS. The VISTA DFCS software design consists of three basic parts: (1) the host DFCS software which controls the aircraft during normal (non-VSS) flight, (2) VSS mode software to accommodate VSS mode transition and VSS input/output, and (3) the VIM software to ensure safe flight during VSS mode operation. The host DFCS software will be modified to accommodate the VSS interface, change the Mux bus architecture, remove the Auto Terrain Avoidance mode, and add any control law changes necessary to maintain the current F-16 handling qualities and stability margins. The VSS mode software permits VSS engagement and disengagement through an engage/disengage request from the VSS, a DFCS Evaluation Pilot Command (DEPC) mode engage request, or a VIM disengage command. All of the VIM functions are implemented in software and will be discussed in further detail in the safety system section.

F-16 Digital Flight Control System

Hardware Description

The F-16 production DFCS uses four identical Digital Flight Control Computers (DFLCC's), that use 1750A instruction set Fairchild 9450 Central Processing Units (CPUs). The F-16 DFLCC's will require some hardware modifications to interface with VSS Hawk #3. The changes will be minimal so that the impact on maintainability is reduced. The Remote Terminal(RT) interface will be modified so that Hawk #3 can interrupt the DFLCC branch CPU that is on the F-Mux. Software maskable synchronization interrupts will be provided between the DFLCC's (see Figure 7). This capability requires minor wiring modifications to the DFLCC motherboard and wiring and microcode changes to the RT sequencer on the Program Memory/1553B RT module. Newly programmed I/O Controller (IOC) file PROMs will replace the production F-16 IOC file PROMs.

VSS/DFCS Communication

The most critical function of the F-Mux is to transmit VSS surface commands from the Hawks to all four branches of the DFCS.[3] The VSS commands fill the F-Mux buffer on the receiving branch. Then the receiving branch will issue an interrupt to the non-receiving branches' CPU's so that the F-Mux data may be transmitted to them. Upon reception of this interrupt, the data will be transferred from the F-Mux buffer to a transmit buffer and then data-linked to buffers in the other three branches. After this data transfer is completed, the branch connected to the F-Mux will issue an interrupt to the CPUs of the branches receiving the data-link transfer. The reception of this interrupt would initiate the processing of the VSS commands. The VIM system checks the data, and if it is ok, sends the commands to the actuators.

In a reverse manner, the DFLCC sends data to VSS Hawk #3. Hawk #3 sends a request for sensor data or surface position output, etc, to the DFLCC branch connected to the F-Mux. This branch passes the request to the other branches, as previously described; however, only the branch connected to the F-Mux sends back data. The data needed by any of the other Hawk computers will be transferred from Hawk #3 over the Hawk Network Interface Card system, which is a parallel-format, high-speed bus linking the Hawks.

Safety System

The objective of the safety system is to provide maximum likelihood of detecting any failures or unsafe conditions, and then revert to the F-16 DFCS. The contract requires that no part of the VSS degrade the reliability of the F-16. The system architecture just described, was designed to meet this objective.

Approach

There are three different 'monitors' consisting of the pilots and two automatic monitors called the VISTA Integrity Management System (VIM). The automatic monitors are the failure detection monitors and the surface and maneuver limit monitors. VISTA safety system concept shown in Figure 8 has these three monitors in its elements.

1. The safety pilot, through his motion cues and displayed information, provides the intelligence for manual disengage in critical non-failure conditions. The safety system ensures that a pilot can obtain DFCS control at all times.

2. The VIM system is based on the System-Wide Integrity Management concept developed and demonstrated on the AFTI/F-16 program and the special safety concepts employed on the current in-flight simulators. Any safety trips in the monitors automatically transfer control to the safety pilot. The VIM operates two types of automatic monitors:

a. The VIM failure monitors are implemented in DFCS and VSS software, and in VSS Engage Logic and Interface Chassis.

b. The VIM surface command and maneuver monitors limit maneuvers to those which will not over-stress or depart the aircraft. They also disengage the VSS if unachievable surface deflections or rates are commanded.

These safety monitors and fail-safe modes are implemented through pilot displays and command modes, DFCS VIM software, and VSS VIM hardware and software.

Pilot Selection of Command Modes

The pilots will monitor the motion cues, visual cues and displays to determine if the aircraft is in danger of pilot induced oscillation, ground impact, or departure. The safety pilot or evaluation pilot can choose to manually disengage the VSS and revert to the safety pilot controlling the aircraft through the F-16 DFCS.

To meet the objective that a pilot can always obtain control of the aircraft through the F-16 DFCS, there are a number of modes that the pilots can engage. They are: the VSS mode, the F-16 primary mode, the DFCS Evaluation Pilot Control (DEPC) mode, and the Digital Backup Unit (DBU) mode. These are shown in Figure 9.

The VSS mode is when the VSS is simulating an aircraft, and all the safety monitors operate. In this mode the VSS computers control the aircraft, and only the Evaluation Pilot's (E.P.) commands are used. The Safety Pilot (S.P.) engages this mode when the E.P. is ready. From this mode all manual and automatic trips put the system into the F-16 primary mode. The F-16 mode uses the basic F-16 DFCS control laws, and is used for taking off and flying to the test points. The S.P. has complete control of the aircraft in F-16 primary mode.

The VSS mode can be disengaged, into the F-16 mode, by the following means. Both cockpits have a thumb switch and a paddle switch on the stick, as well as a switch on the throttle. There is a switch on the top left instrument panel, near the VSS panels in both cockpits. The front cockpit also has a switch on the center stick. In addition to these switches, if the S.P. yanks on the stick, giving a pitch or roll force input above a set level, the system will revert from the VSS to the F-16 primary mode.

If the safety pilot should want the evaluation pilot to fly the aircraft in a pilot relief mode, or if the S.P. should become incapacitated unexpectedly, the E.P. could engage the DEPC mode. The E.P. would have control of the aircraft directly through the DFCS. The S.P. can regain control of the aircraft by depressing the paddle switch on the grip. The last mode, the DBU mode, can be selected from any other mode, by the S.P. The DBU mode automatically engages if the primary DFLCC branch fails. In this mode the E.P. and S.P. commands are summed, but the S.P. can win a force fight by depressing the paddle switch. These modes retain the reliability of the F-16 DFCS, and ensure that a pilot can obtain control in any circumstance.

DFCS VIM Functions

Along with pilot monitoring and manual trips there are automatic monitoring and trips in the VIM. The VIM monitor signals are all combined in the 'VIM Control' software module shown in Figure 10. This module combines signals from the VIM monitors in the DFCS software, the VSS Engage Chassis, and the VSS Hawk computers. It then sends a VSS disengage request to the VSS engage and disengage logic software module when there is a safety trip. The software monitors in each of the DFLCC's are further described.

Communications Failure Monitors

The VIM determines the health of the VSS by monitoring F-Mux data. A checksum monitor will have the Hawk computers add the values of the surface commands and transmit the sum. The DFLCC's will repeat the addition and compare its sum with that of the Hawk's. This goes beyond normal parity checking and can detect double bit errors. A heartbeat monitor will test that a VSS 'happy signal' toggles back and forth with every new data package. Since the VSS surface commands are sent to the asynchronous DFLCC's via interrupts, the DFLCC's are not looking for the data on a regular basis. A transmission rate monitor checks that the new data is coming in at the rate that the VSS said it would. The VIM also monitors the B-Mux to ensure that the Inertial Navigation Unit (INU) data is being updated regularly.

Surface Command Monitors

The 'dual command' monitors check that the differences between the separately generated surface commands coming from the VSS agree and are within position limits. The rate of command is then checked. The structural position limit monitor limits the flaperon in a manner similar to the AFTI/F-16 flaperon structural limiter. The F-16 was designed for symmetric flap deflections during takeoff and landing, not during high-speed maneuvering flight.

The pitch, roll and yaw monitors compare VSS surface commands with F-16 control law generated surface commands.[4] Figure 11 shows the VIM pitch monitor, which is typical of the three. When the two 'VSS engage' switches close, the F-16 pitch axis control laws are forced to track the VSS surface commands by adjusting the pilot stick commands. This has the advantage of initializing the F-16 control laws for reversion from VSS to DFCS control. The mixer combines the symmetric pitch deflection and antisymmetric roll deflection for each horizontal tail, and visa-versa for the demixer.

The F-16 pitch control laws have an angle of attack (alpha) limiter. This limits alpha as a function of roll rate, pitch rate and alpha. This limiter as well as others in the pitch control law protect against departing the aircraft. The axis monitors try to use this protection in the VSS system by limiting the difference between the VSS commands and the limited F-16 commands.

In the case of the pitch axis monitor, the error signal between the VSS and DFCS elevator commands are multiplied by a constant to predict the commanded load factor (g's). Figure 12a shows a simple test case of a hard-over VSS command, which would produce a load factor of over 9 g's, as shown in Figure 12b. Figure 12a shows the hard-over VSS surface command of -20 deg., the actuator rate limited command approaching -20 deg. The F-16 control law generated command approaches -20 deg. and then backs off as the alpha limiter starts to take effect. Figure 12b shows how the scaled error signal (pitch monitor output) detects the hard-over. The example problem was done without any other monitors working, such as the command rate limiter. The roll and yaw monitors work similarly. These monitors provide the extra level of failure detection needed in a single thread system such as the VSS.

The feedback structural monitors examine alpha, g's, and roll rate to make sure the aircraft does not exceed the limits based on structural load restrictions. These restrictions are a function of stores loading.

Maneuver Restrictors

The VISTA will operate within 80-90% of the F-16 flight envelope. Operational Mission Restrictors (OMR) monitors insure that VISTA does not exceed limits in normal load factor, lateral acceleration, angle of attack, angle of sideslip, and Mach. The software will have hard upper limits, and settable reduced upper limits which can be reset by the safety pilot for particular missions.

VSS VIM Functions

Dual Sensor Signal Monitors

The Hawk computers will receive two independent sensor signals. Hawks #2 and #3 will each compare the signals and check that they are in range. If the sensors are determined to be healthy, the average of the signals is used. The dual signals from the new accelerometers, rate gyros, sideslip probes and centerstick force transducers will be compared.

ASE Detection

A monitor is being designed which will determine if there is an unstable coupling of the flight control system, structure and aerodynamics. Current plans are to implement a Fast Fourier Transform (FFT) type analysis of the frequency content of the flight control commands. If a rapid onset of energy at a particular frequency of interest is detected, VISTA will revert back to the DFCS. Seven accelerometers on the wingtip, horizontal tail, and vertical tail will be monitored by telemetry during ASE flight tests.

Hardware Monitors

The Engage Logic and Interface Chassis monitors the VSS hardware through dual analog safety trip circuits. It monitors the feel system servos, including the rear cockpit throttle servo. The Hawk and Titan computers send discrete disengage signals from their own internal monitors as well as watchdog timer discrete signals. These signals are 'or'ed together so that any one trip signal will disengage the VSS. Dual discrete disengage signals are sent to the DFLCC's upon a safety trip.

Acknowledgement

This paper is derived from work done on 'Variable Stability In-Flight Simulator Test Aircraft (VISTA) Development Program', USAF contract number F33615-88-C-3611. The authors wish to acknowledge the help in preparing this paper provided by engineers at General Dynamics and Calspan.

Conclusion

The VISTA/F-16 aircraft is a new high performance in-flight simulator that provides improved simulation capabilities and performance envelope. VISTA interfaces four new VSS computers with the existing F-16 DFCS through a unique mux-bus architecture. Safety in this single-thread system has been assured through multiple reversion paths to the fail-safe condition and multiple safety monitors. All possible VSS failures will result in control being transferred back to the F-16 DFCS, without any degradation in the basic F-16 reliability and performance.

References

[1] G. K. Hellmann, D. E. Frearson, J. Barry, Jr., 'VISTA/F-16: The Next High-Performance In-Flight Simulator,' Flight Simulation Technologies Conference, Atlanta, GA, September, 1988.

[2] M. V. Dunbar, G. K. Hellmann, 'VISTA/F-16: An In-Flight Research Tool,' National Aerospace Electronics Conference, Dayton, OH, May, 1989.

[3] General Dynamics,'Class II-1 Modification Document for VISTA', Preliminary Document, March, 1989.

[4] General Dynamics, 'Preliminary Design Review Viewgraphs,' VISTA PDR, April, 1989.

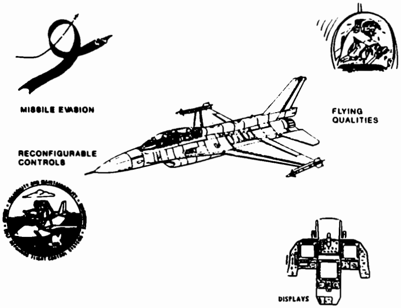


Fig. 1 Possible Vista Applications

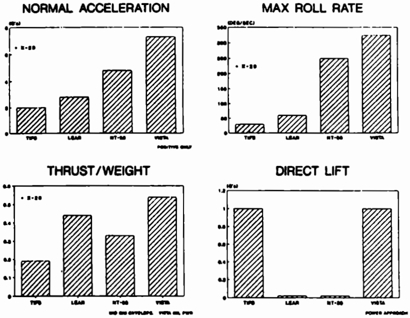


Fig. 2 Comparison of Simulator Performance

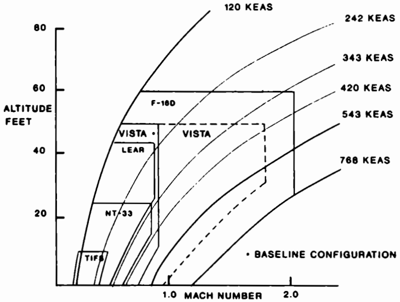


Fig. 3 U.S. Simulators Performance Envelopes

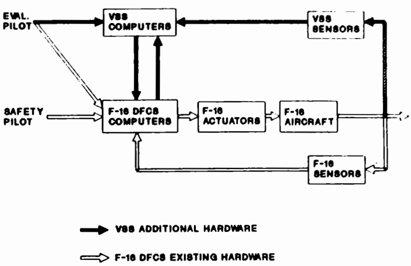


Fig. 4 VISTA Architecture

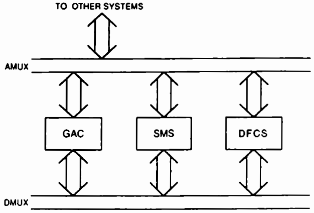


Fig. 5a Production F-16 Mux Architecture

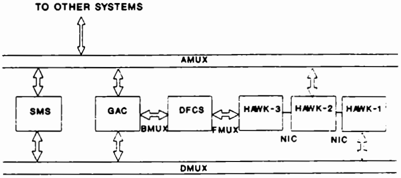


Fig. 5b VISTA/F-16 Mux Architecture

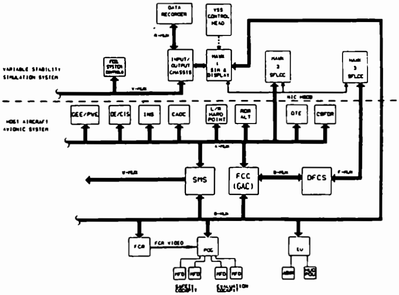


Fig. 6 VSS/Host Aircraft Avionic Systems

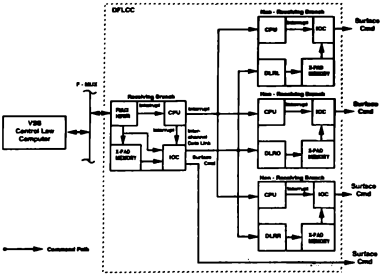


Fig. 7 DFCS/F-Mux Interface

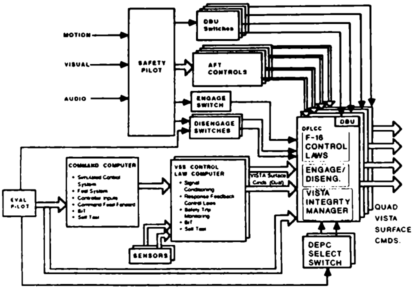


Fig. 8 Safety System Features

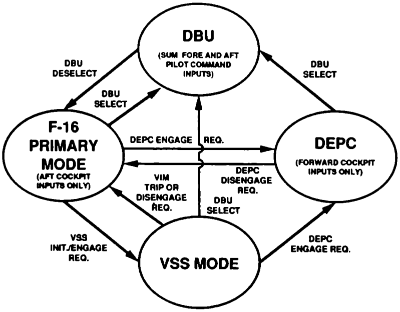


Fig. 9 VISTA DFCS Mode Diagram

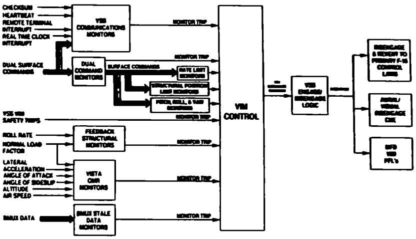


Fig. 10 DFCS VIM Functional Diagram

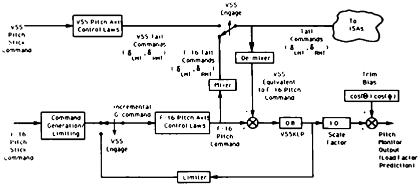

VSS Pitch Stick Command
VSS Pitch Axis Control Laws
VSS Tail Commands (δ_{LHT} , δ_{RHT})
VSS Engage
To ISAs
Tail Commands (δ_{LHT} , δ_{RHT})
F-16 Tail Commands (δ_{LHT} , δ_{RHT})
De-mixer
Mixer
VSS Equivalent to F-16 Pitch Command
Trim Bias
cos(Θ) cos(ϕ)
F-16 Pitch Stick Command
Command Generation/ Limiting
Incremental G command
VSS Engage
F-16 Pitch Axis Control Laws
F-16 Pitch Command
0.8
VSSKLP
1.0
Scale Factor
Pitch Monitor Output (Load Factor Prediction)
Limiter

Fig. 11 Pitch Monitor

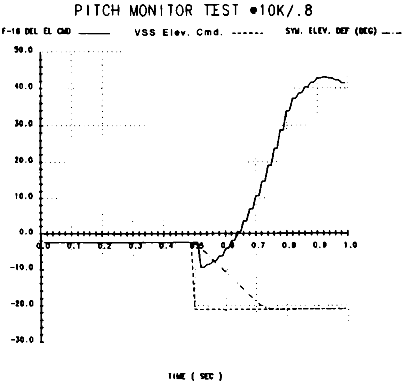


Fig. 12a Pitch Monitor Example - Input

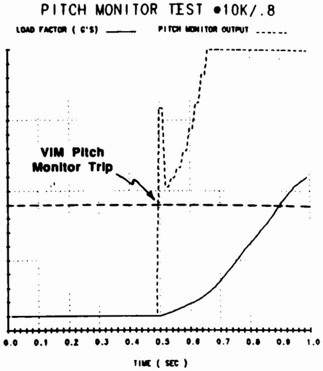


Fig. 12b Pitch Monitor Example - Output

89-3331-CP

MAPPING LABORATORY TESTS TO IN-FLIGHT TASKS[1]

Valerie J. Gawron[2]
Louis H. Knotts[3]
Calspan Advanced Technology Center
Buffalo, New York

Samuel G. Schiflett
School of Aerospace Medicine

ABSTRACT

Much research is being conducting to identify the effects of drugs on pilot performance. Some of this research is conducted in the laboratory, some in-flight. Often the results of the laboratory and in-flight studies are not the same. The sources of these differences can include: different subjects, additional stressor effects of flight, and an inadequate mapping of the laboratory and flight tasks. The first source can be eliminated by a strong experimental design. The effects of the second source can be interpolated from multi-stressor laboratory experiments. The third source can be eliminated by using a special set of laboratory tasks and a validated mapping scheme. Both of these are described in the following paper.

INTRODUCTION

Much laboratory research has been conducted recently to identify the effects of a wide range of drugs on performance. Results from these studies are then used to guide the design of in-flight research examining the same effects in-flight. But how can a researcher extrapolate results from simple laboratory tasks, e.g., single-axis tracking, to the complex tasks of controlling an aircraft. As part of a inflight evaluation of pyridostigmine bromide (PB), Calspan developed a methodology to do just that:

Step 1 - Review Laboratory Research

Step 2 - List the Laboratory Tasks and the Effects of Drugs on Each Task

Step 3 - Map the Tasks to the Unified Tri-Service Cognitive Performance Assessment Battery (UTC-PAB)

Step 4 - Identify Aircraft and Mission Characteristics

Step 5 - Map UTC-PAB Tasks to the Aircraft and Mission

We began by reviewing the laboratory studies on PB.

Step 1 - Review Laboratory Research

Graham and Cook (1984) examined the effects of PB on performance of a multiple-task battery in a controlled laboratory environment. They reported performance decrements associated with PB in single-task performance of a probability-monitoring task and in dual-task performance of a memory-search task with visual tracking. They also found performance enhancements: depth perception accuracy (matching the distance of two placards) was improved about 3 mm; hand steadiness was increased; and visual-contrast sensitivity improved at 3 cycles per degree (c/d). It is noteworthy that for other c/d grids (0.5 through 22.8 c/d), no differences in performance occurred.

In a similar study by Kay and Morrison (1985), subjects performed a visual-contrast task 1.5 to 2.23 hr after ingesting PB. The authors found no performance enhancement even at 3 c/d. Finally, Borland, Brennan, Nicholson, and Smith (1985) found no effect of PB on contrast sensitivity; however, the frequencies they evaluated did not include 3 c/d.

Borland, et al. (1985) subjected four young men to a multitude of performance tasks after doses of PB. The authors reported two performance enhancements associated with PB: 1) mean critical flicker fusion was raised and 2) fewer responses were missed on a dynamic visual-acuity test (identifying the location of the break, in a Landolt C projected onto a rotating mirror). The overall accuracy of the acuity test did not differ from the placebo condition, however. Further, these researchers found no effects on the following: digit symbol substitution, symbol copying, pupil diameter, macular threshold (pupillary response to light flashes), or kinetic quantitative perimetry (ocular response to varying illumination levels). As in the Graham and Cook study, visual-tracking performance was degraded.

Schiflett, Stranges, Slater, and Jackson (1987) examined the effects of PB in combination with mild hypoxia. Each of their subjects performed a series of tests that measured sensory, motor, and cognitive functioning at ground level and at altitudes of 8,000 and 13,000 ft. PB did not alter performance.

Step 2 - List the Laboratory Tasks and the Effects of Drugs on Each Task

The effects of PB on the performance of laboratory tasks are summarized in Table 1.

[1] This research was funded by the Joint Working Group on Drug Dependent Degradation in Military Perofrmance, Army Medical R&D Command, Fort Detrick, MD.

[2] Human Factors Engineer, Flight Research Department, Member AIAA.

[3] Principal Aeronautical Engineer, Flight Research Department, Member AIAA.

Table 1. Summary of Laboratory Research

Laboratory Task	Effect of PB
Probability-Monitoring	Degrade Performance
Memory Search with Visual Tracking	Degrade Performance
Depth Perception	Increase Accuracy
Hand Steadiness	Enhance Performance
Visual Contrast	Increased Sensitivity
Critical Flicker Fusion	Degrade Performance
Dynamic Visual Acuity	Decrease Number of Misses
Digit Symbol Substitution	No Effect
Symbol Copying	No Effect
Kinetic Quantitative Perimetry	No Effect
Visual Tracking	Degrade Performance

Step 3 - Map the Tasks to the UTC-PAB

To ensure that the effects of drugs are investigated for the full spectrum of human skills, the UTC-PAB (Perez, Masline, Ramsey, and Urban, 1987) was developed. It has 25 tasks organized in 6 areas (see Table 2). Each task is described below. The mapping of laboratory tasks to UTC-PAB tasks is presented in Table 3.

Table 2. UTC-PAB Organization Scheme

I. Perceptual input, Detection, and Identification
   - Visual Scanning Task
   - Visual Probability Monitoring Task
   - Pattern Comparison (Simultaneous)
   - Four-Choice Serial Reaction Time

II. Central Processing
   - Auditory Memory Search (Memory Search Tasks)
   - Continuous Recognition Task
   - Code Substitution Task
   - Visual Memory Search (Memory Search Tasks)
   - Item Order Test

III. Information Integration/Manipulation--Linguistic/Symbolic
   - Linguistic Processing Task
   - Two-Column Addition
   - Grammatical Reasoning (Symbolic)
   - Mathematical Processing Task
   - Grammatical Reasoning (Traditional)

IV. Information Integration/Manipulation—Spatial Mode
   - Spatial Processing Task
   - Matching to Sample
   - Time Wall
   - Matrix Rotation Task (Spatial Processing Task)
   - Manikin Test
   - Pattern Comparison (Successive)

V. Output/Response Execution
   - Interval Production Task
   - Unstable Tracking Task

VI. Selective/Divided Attention
   - Dichotic Listening Task
   - Memory Search Unstable Tracking Combination (Sternberg-Tracking Combination)
   - Stroop Test

Perez, et al. (1987, p. 11)

1. Linguistic Processing Task "The purpose of the Linguistic Processing Task is to test a subject's ability to code linguistic information at different depths of processing. The task places variable demands upon the resources associated with the processing and transformation of linguistic information." (Perez, et al. 1987, p. 14)

2. Grammatical Reasoning (Traditional) "The purpose of the grammatical reasoning test is to measure the subject's general reasoning ability. This test is a type of sentence verification task that taps the processing capacity of working memory. Furthermore, it is known to be sensitive to environmental stress, pollutants, and the effects of sleep loss." (Perez, et al. 1987, p. 24). The traditional grammatical reasoning task has been used to examine nitrogen narcoses (Baddeley, 1968), traffic pollution (Lewis, Baddeley, Bonham, and Lovett, 1970), hypothermia (Baddeley, Cuccaro, Egstrom, Weltman, and Willis, 1975), and sleep loss (Angus and Heslegrave, 1985).

3. Grammatical Reasoning (Symbolic) "The purpose of this task is to tap resources dedicated to general reasoning ability. The symbolic grammatical reasoning task is a type of sentence verification task that taps the processing capacity of working memory. This task is known to be sensitive to variable information processing demands and is probably sensitive to environmental stress, pollutants, and sleep loss." (Perez, et al., 1987, p. 34). Grammatical reasoning tasks have been used to investigate nitrogen narcosis (Baddeley, 1968), traffic pollution (Lewis, Baddeley, Bonham, and Lovett, 1970), hyperthermia (Baddeley, Cuccaro, Egstrom, Weltman, and Willis, 1975), durinal variations (Folkard, 1975), and sleep loss (Poulton, Hunt, Carpenter, and Edwards, 1978).

4. Two-Column Addition "The purpose of this subject-paced, mental arithmetic test is to measure the subject's ability to sum simple addition problems. The test is diagnostic of the speed and accuracy with which subjects retrieve arithmetic information (e.g., math facts) and utilize procedural knowledge (e.g., well learned procedures for adding columns of digits). In addition, short term storage of carry and intermediate result information is required." (Perez, et al., 1987, p. 51). Mental addition tasks have been used to examine the effects of carbon monoxide (Johnson, Cohen, Struble, Setzer, Angur, Gutuik, McDonough, and Hauser, 1974), norganic lead (Repko, Morgan, and Nicholson, 1975), methly chloride (Repko, Jones, Garcia, Schneider, Roseman, and Corum, 1976), atropine, ditran, and scopolamine (Ketchum, Sidell, Crowell, Aghajanian, and Hayes, 1973), and parpanite (Michelson, 1961).

5. Mathematical Processing Task "The purpose of this self-paced mental arithmetic task is to test a subject's information processing resources associated with working memory. Specifically, the subject is required to: (a) retrieve information from long term memory, (b) update information in working memory, (c) sequentially execute different arthmetic operations, and (d) perform numeric comparisons." (Perez, et al., 1987, p. 61)

6. Continuous Recognition Task "The Continuous Recognition Task is designed to place variable demands upon processing resources associated with encoding and storage in working memory. The task tests a subject's ability to encode, reherse, recall, and compare numbers in short term memory on a continuous basis." (Perez, et al., 1987, p. 75)

7. Four Choice Serial Reaction Time "This task is designed to evaluate information processing resources dedicated to stimulus encoding and categorization, and response selection, although it is probable that resources dedicated to encoding are tapped most heavily." (Perez, et al., 1987, p. 87)

8. Visual Memory Search Task "The purpose of the Alpha-Numeric Visual Vigilance Task (ANVVT) is to test a subject's ability to continue making decisions and rapid responses to visual symbols for long nonstop periods. The ANVVT is a discrimination reaction task intended to simulate a situation in which a person monitoring a visual display might show fatigue and performance decrement without being aware of it." (Perez, et al., 1987 p. 101)

9. Memory Search Tasks "The purpose of this memory search task is to test a subject's ability to make comparisons of letters maintained in memory. The task is diagnostic of the processes of selective retrieval and comparison in short term working memory. This task may also reflect processes involved in the encoding of stimulus items, categorization, response selection, and response execution." (Perez, et al., 1987, p. 110)

10. Spatial Processing Task "This task is designed to examine the subject's ability to mentally rotate a series of histograms prior to making a same/different judgment about them. The task taps visual short term memory, since the standard and test stimuli are presented successively rather than simultaneously." (Perez, et al., 1987, p. 136)

11. Matrix Rotation Task "The purpose of the Matrix Rotation Task is to assess the subject's facility for spatial rotation. Spatial rotation, also known as spatial transformation, is one component of spatial orientation. This task also evaluates short term perceptual memory." (Perez, et al., 1987, p. 146)

12. Manikin Test "The purpose of the Manikin Test is to assess the subject's ability to perform rotations and related transformations of a mental image. This ability is one of the three general subdivisions of spatial ability. Lohman (1979) has called this ability spatial orientation (SO), which requires mental movement of the self to view the test stimulus from a new perspective." (Perez, et al., 1987, p. 155).

13. Pattern Comparison (Simultaneous) "The primary purpose of this self paced pattern comparison test is to assess the subject's perceptual speed. Perceptual speed is one aspect of general spatial ability. The test provides information about the subject's ability to make simultaneous judgments about the similarity of two patterns." Perez et al., 1987, p. 164).

14. Pattern Comparison (Successive) "The primary purpose of this task is to examine the subject's short term spatial memory and perceptual speed. The test is diagnostic of spatial memory, since the subject must maintain the standard in memory while the comparison with the test pattern is being made." (Perez, et al., 1987, p. 174)

15. Visual Scanning Task "This task is a modification of Neisser's (1963) letter search task which requires subjects to search for and detect a target embedded in nontarget items. This test is diagnostic of a subject's ability to perform rapid visual pattern discrimination." (Perez, et al., 1987, p. 184)

16. Code Substitution Task "This task is designed to tap information processing resources dedicated to the rapid encoding and associative evaluation of stimuli." (Perez, et al., 1987, p. 198)

17. Visual Probability Monitoring Task "The purpose of this task is to test perceptual resources devoted to scanning and detecting of visual signals." (Perez, et al., 1987 p. 205)

18. Time Wall "The purpose of the time wall task is to test a subject's ability to estimate the time at which a target, moving at a constant rate, will have traveled a predetermined distance. That is, on each trial the subject must integrate the available speed and distance information in order to correctly anticipate the time at which the target reaches a certain spot on the screen." (Perez, et al., 1987, p. 218)

19. Interval Production Task "This task was designed to be used as a secondary task to measure demands placed on motor output by a primary task (Michon, 1966). However, it may be used as a stand alone test to examine the degree to which variables such as drugs, environmental stress, and toxic substances distrupt manual response timing." (Perez, et al., 1987, p. 230)

20. Stroop Test "This test is a modified version of the classic color-word test developed by Stroop (1935). The purpose of this test is to measure a subject's susceptibility to response interference." (Perez, et al., 1987, p. 239)

21. Dichotic Listening Task "This test evaluates information processing resources dedicated to auditory selective attention." (Perez, et al., 1987, p. 250)

22. Unstable Tracking Task "This task tests information processing resources dedicated to the execution of rapid and accurate manual responses." (Perez, et al., 1987, p. 263)

23. Memory Search-Unstable Tracking Combination "This dual task combination is intended to tap information processing resources dedicated to time sharing ability; that is, the ability to perform two tasks concurrently." (Perez, et al., 1987, p. 278)

24. Matching to Sample "This task is designed to assess the subject's ability to quickly and accurately choose a test stimulus which is identical to a standard stimulus presented previously. The test taps short term spatial memory and pattern recognition skills." (Perez, et al., 1987, p. 288)

25. Item-Order Test "The purpose of the item-order test is to examine a subject's ability to recognize strings of letters as being the same or different. Error rates produced from this test should reflect processes of short term memory recognition." (Perez, et al., 1987, p. 297)

Table 3. Mapping Laboratory Tasks to UTC-PAB Tasks

Laboratory Task	UTC-PAB Task
Probability-Monitoring	Visual Probability Monitoring
Memory Search with Visual Tracking	Memory Search Unstable Tracking Combintion
Depth Perception	Matrix Rotation
Hand Steadiness	Unstable Tracking
Visual Contrast	Pattern Comparison
Critical Flicker Fusion	Pattern Comparison
Dynamic Visual Acuity	Unstable Tracking Task
Digit Symbol Substitution	Code Substitution
Symbol Copying	Grammatical Reasoning (Symbolic)
Kinetic Quantitative Perimetry	Unstable Tracking
Visual Tracking	Unstable Tracking

Step 4 - Identify Aircraft and Mission Characteristics

Both the Air Force and Army were concerned with the effect of PB on C-130 aircrews to perform an airdrop mission. Given the immense safety considerations both to the subjects and to ground personnel, the USAF Flight Dynamics Laboratory Total In-Flight Simulator (TIFS) was used in this study. The TIFS (Figure 1) is a research aircraft developed by Calspan Corp., Buffalo, NY, for use in flight-testing advanced flight control technology, avionics, and cockpit instrumentation. The basic aircraft is a C-131H (Convair 580) which has been modified for use as an in-flight simulator. The TIFS has direct-lift flaps, side-force surfaces, and a variable-stability system. These features allow six degree-of-freedom dynamic simulation in flight and replication of the flying characteristics of other flight vehicles. TIFS has two cockpits: a normal C-131 cockpit occupied by two safety pilots and a simulation cockpit occupied by two subjects.

Flight Safety. TIFS is always under the command of the safety pilots even when flown by the subjects in the simulation cockpit. The controls in the safety cockpit are mechanically connected to the aircraft's control surfaces and always indicate the motion of each surface. A single button in the safety cockpit disengages the variable-stability system and leaves the safety pilots with direct mechanical control of the aircraft. In addition, automatic monitor circuits have been programmed to prevent extreme signal inputs to the control surfaces from the simulator cockpit and to prevent inadvertantly exceeding aircraft structural limits.

TIFS Simulation of the C-130. For this experiment, the simulation cockpit was modified to resemble a C-130 tactical-transport aircraft. In addition, aircraft handling qualities similar to the C-130 were generated by the variable-stability system.

Mission Profile. The mission profile developed for this study was a two-hour low altitude heavy-equipment extraction. The mission profile is described in Table 4.

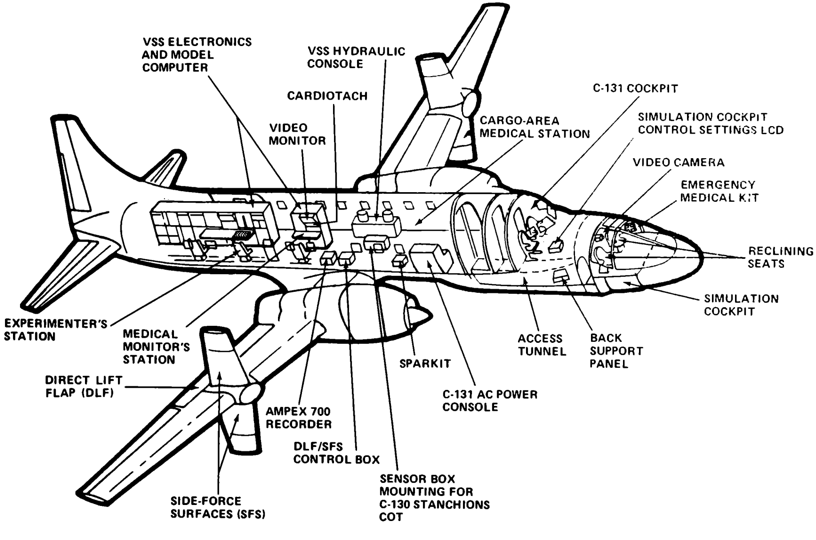


Figure 1. TIFS

Table 4. Description of the Mission Profile

Minutes	Maneuvers
0-1	CLIMB: Climb and turn on course. Objective scoring of airspeed (170 KIAS), vertical velocity (1000 fpm), and bank angle (0 degrees).
1-37	INGRESS: Simulated formation flight with station-keeping equipment (SKE). Objective scoring of altitude (1000 ft above highest obstacle within 5 NM), airspeed (210 KIAS), bank angle (0 degrees), waypoint arrival, and formation position.
37-47	DROP: Slowdown and descent to drop altitude. Simulated air drop at 1100 ft above ground level (AGL). Escape (turn and climb). Objective scoring of airspeed (130 KIAS), altitude (1100 ft AGL), bank angle (20 degrees), and simulated air drop score.
47-65	EGRESS: Simulated formation flight with station keeping equipment (SKE). Objective scoring of altitude (1000 ft AGL), airspeed (210 KIAS), bank angle (0 degrees), waypoint arrival, and formation position.
65-75	APPROACH: ILS approach and landing. Objective scoring of airspeed assigned by Air Traffic Control (ATC), vertical velocity (1000 fpm) and bank angle (0 degrees), localizer and glideslope maintenance, and yoke and throttle movements.

During ingress, the main task for both the pilot and copilot was station keeping. The device presents range information in the form of concentric circles, each representing 4000-ft separation. An airspeed of 210 +/- 20 KIAS was simulated for the two-ship formation. The copilot also performed low-level navigation and communication, and responded to checklist calls from the pilot. The slowdown began at the entry point, approximately 16 NM from the drop zone. At the stabilization point, the aircraft was 6 NM from the drop zone and was on drop altitude (1100 AGL), drop heading, and drop airspeed. At this point, the simulator aircrew transitioned from an SKE drop profile to visual airdrop procedures: they were given simulated surface winds which required adjustments to accomplish the visual drop. When the red light came on, the simulator aircrew transitioned from visual airdrop procedures back to SKE drop procedures. Cruise legs were flown at 210 KIAS. The drop was flown at 130 KIAS. A transient ramp pitching moment was used to simulate the extraction of 4000 lb of heavy equipment. During the escape leg, the pilot descended, accelerated, turned, and completed the after-drop checklist.

Step 5 - Map UTC-PAB Tasks to the Aircraft and Mission

Flight tasks can created to match the sensory, motor, and cognitive resource demands of the UTC-PAB tasks as closely as possible but still maintain the validity in an operational environment. Examples from the PB study are given below.

Responding to an Engine Fire - A simulated engine fire occurred during the egress segment of one of the four data flights. The data flight was randomly selected. The fire was scheduled to begin at a randomly selected time between 2 and 20 min after the level off. The fire was simulated by illumination of the master fire warning (MFW), a steady fire light in one of the four fire handles mounted above the windscreen, and the sounding of a warning ("whoop") horn. The correct response was for the pilot to give the command, "Condition lever," followed by the number of the engine (one, two, three, or four) and "feather." At the pilot's command, the copilot was required to move the appropriate lever in the center console to the full-back position. The pilot then pulled the corresponding fire handle and stated, "Engine," followed by the number of the engine on fire (one, two, three, or four) and "Fire handle pull." When he pulled the fire handle, the simulator copilot discharged the fire extinguishing agent. Finally, the simulator pilot called for the cleanup checklist. At that time, a safety pilot informed the simulator crew: "Emergency terminated."

Completing Checklists - Nine checklists were scheduled to be completed during each familiarization or data flight at the following times: 1) before flight, 2) after takeoff, 3) 20 minute, 4) 10 minute, 5) slowdown, 6) 1 minute, 7) drop, 8) descent, and 9) before landing.

Detecting an Oil-Pressure Change - A change in oil pressure in one of the four engines occurred during the ingress segment of the mission. The change was initiated at a randomly selected time, from 1 to 36 minutes after level-off. The random selection was to ensure a full range of values for the start time. This segment was chosen because the crew would be concentrating on navigating to the drop point rather than on monitoring engine status. The engine that had the error was randomly selected, with one constraint: that each oil-pressure gauge fail an equal number of times across the entire experiment. Because of the 360-degree face of the dial, the error was 3 degrees/minute with a maximum error of 20 degrees. If the error was not detected, it continued throughout the remainder of the flight.

Detecting a Heading Discrepancy - An incipient heading-angle error (6 degrees/min) was introduced into both the pilot's and copilot's Horizontal Situation Indicators (HSI) during the egress segment of the data flights. This error could result in a maximum 15-degree discrepancy in heading angle between the HSI and the radio magnetic indicator (RMI). The error began at a time selected randomly between 1 and 10 minutes after the start of the segment. The selection was pseudorandom because of three constraints: 1) each time assignment must occur at least four times, and no more than five, across the entire experiment, 2) the heading error could not start simultaneously with the engine fire, and 3) the four start times for each crew had to be unique. The error was corrected automatically if not detected before the beginning of the next segment, i.e., illuminated landing system (ILS) approach.

Controlling the Aircraft - When the landing gear was raised, the simulation pilot controlled the aircraft by using the fully functioning yoke, rudders, and throttles. Altitude, airspeed, vertical velocity, bank angle, yoke and throttle movements, and flap settings were recorded every two hundredths of a second during every segment of the mission. Localizer and glideslope error were recorded during the ILS approach and landing segment. Throughout the familiarization (fam) and data flights, the simulator pilot also instructed the simulator copilot to set flaps and to raise and lower the landing gear. Flap settings varied by mission segment: 0% during ingress and the ILS approach: 50% during the air drop and egress segments; and 100% during landing.

Of special interest was the simulator pilot's performance during a 3-degree step change in pitch during the simulated air drops. This change was created using TIFS' variable-stability feature and occurred automatically 1 second after the chute-release pushbutton was pressed. Note that the chute-release pushbutton was armed by the onboard technician during the slowdown, and disarmed after the simulated drop, to avoid the danger of accidental activation of the ramp change during another segment of the flight. The chute-release pushbutton was armed even if a "No drop" signal were given in the experiment, thus allowing the aircrew to drop heavy equipment in error.

The pitch change was a ramp increase in pitch (to simulate heavy equipment moving aft of the center of gravity) followed by a 3-degree stepdown in pitch (to simulate the equipment leaving the aircraft). The 3-degree, pitch-down attitude was selected to provide the simulator pilot with a difficult task to perform during the low airspeed (130 knots) nose-up attitude. This set of conditions kept the TIFS just above stall conditions in a C-130 type aircraft.

Station Keeping - The station-keeping task simulated formation flying and required pursuit tracking of a mythical lead aircraft. The pilot was instructed to maintain 4000-foot spacing between TIFS and the lead aircraft, using simulated SKE. The display presented a set of 4000-ft range rings, with the lead aircraft in the center. The position of the lead aircraft was marked on the display with a box symbol in the 12 o'clock position. If the lead aircraft symbol was drifting downward on the display, the response was to decrease airspeed; upward drift required speeding up. The pilot's task was to match the lead aircraft's flight path.

Aerial Navigation - The copilot was responsible of aerial navigation during all segments of the mission. His job was aided by a computer flight plan generated by a C-130 navigator before each flight. The plan identified the estimated time of arrival (ETA) at waypoints A through F. Used as data were deviations from these ETAs, and the distance from the waypoints at the simulator crew's waypoint back.

Authentication - The copilot received an authenticate command (e.g., "Authenticate Bravo Zulu") from a safety pilot during the ingress segment of the familiarization and data flights, just after the slowdown (start of the drop segment). The copilot scanned his authentication code book to find the page with the appropriate date and time. When the copilot found the appropriate page, he read back to the safety pilot the two letters identified by the column heading from the first letter in the commnd (e.g., Bravo) and the row heading from the second letter in the command (e.g., Zulu). The letters were selected randomly, with the constraint that each letter occur at least once during the entire experiment.

Changing Radio Frequencies - During the preflight planning, the simulator aircrews were given a list of six VHF frequencies identified as TAC A, B, C, D, E, and F. A separate set of frequencies was assigned for use during each data flight. The frequencies were randomly selected but with three constraints: 1) each frequency occurred at least twice but never more than three time across all data flights, 2) all six frequencies within a set were unique, and 3) special use frequencies (e.g., 120.5, 121.5, 121.9, 1122.6, and 123.9) were not used.

The copilot switched from one to another of the six frequencies in response to radio calls from a simulated formation-lead aircraft (for example, "Go to TAC A"). For this experiment, these calls were initiated by a safety pilot shortly before the 10-minute checklist, during the slowdown, and at the start of the egress segment. The simulator copilot responded by tuning in the VHF frequency corresponding to the TAC frequency, and transmitting the call sign followed by the new frequency designation. Call signs were randomly selected English words with three to five letters. Each word was followed by a two-digit number. Each digit (zero to nine) occurred at least 9 times and no more than 10 times throughout the experiment.

Returning from an Empty Channel - During one of the frequency changes, the new frequency was an empty channel. Hence, the simulator copilot had to return to the previous frequency, ask for an operational channel, enter it into the VHF head, and confirm that the channel was indeed operational by transmitting a message. If the simulator pilot failed to change back to the original frequency within 5 minutes, a safety pilot would command the simulator copilot to go to the new operational frequency.

Responding to an IFF Query - Once during ingress and once during egress, a safety pilot gave an Identify Friend or Foe (IFF) query: for example, stated "One, two, three, four, IDENT." The simulator copilot then moved the four IFF thumbwheels to the settings given in the IFF request, and pressed the IDENT pushbutton. Each IFF thumbwheel could be set to a single digit from 0 through 9. Time for the IFF query was random, with two constraints: 1) the query during the egress segment could not happen simultaneously with the engine fire and 2) each start time had to be unique across all four data flights for a single crew. Numbers for the IFF query were randomly selected with one constraint: all digits had to occur at least six and no more than seven times, across the entire experiment.

Updating the Release Point - After responding to the predrop authentication command, the simulator copilot received winds at surface information from drop zone personnel (simulated by a safety pilot). This information could be, for example, "Winds at surface 180 degrees, 10 knots." The winds at surface vlaues were pseudorandomly selected to be +/-20, +/-25, +/-30, +/-35, or +/-40 knots different from the winds presented at the preflight briefing. There was one constraint on selection of the wide deviation: that each deviation occur an equal number of times in the drop and no-drop data flights.

The wind direction identified which radius on the windface was to be used. The copilot used this information to update the computed air release point (CARP), calculated during the preflight briefing, by using a transparent circular windface. The center of the windface was overlaid on a scaled aerial photograph of the drop zone. The center was displaced by the Forward Travel Distance (FTD) presented at the preflight briefing. Wind velocity, assigned altitude, and initial heading from the initial point (IP) were presented at the preflight briefing, and remained unchanged.

Dropping Heavy Equipment - Immediately before the simulated air drop, the simulator copilot monitored two voice channels: the ground personnel (represented by the onboard experimenter). Unless the copilot heard a "No drop" signal from either channel, he pressed the "chute-release" pushbutton in the center console of the simulation cockpit, and moved the jump-signal toggle on the right bulkhead to the "go" position. The latter action caused the green drop-status light to illuminate. Note that pressing the chute-release pushbutton caused the equipment to be dropped even if the jump signals had not been moved to the go position.

A no-drop signal required only that the copilot move his hand away from the chute-release pushbutton. Subjects were briefed that air drops occurred in three out of four opportunities in the operational environment, and that they could expect similar probabilities in this experiment. In reality the simulated air drops occurred in two out of four opportunities. The order of the drop or no-drop signals was random across data flights, with one constraint: two drops and two no-drops occurred for each aircrew.

Responding to Pilot Commands - The copilot adjusted flaps and raised and lowered the landing gear upon command from the pilot.

Secondary Task - Estimates of inflight workload were obtained from performance of a self-paced, secondary task: specifically, the Sternberg memory search task (task 9 of UTC-PAB). Single letters were presented in the lower left-hand area of both the pilot's and the copilot's SKE display, with separate letters being presented to each. Both the pilot and copilot were instructed to activate a slider switch on their yokes when a letter was presented. They were told to push the slider upward if the letter belonged to one of eight sets of four letters memorized before the flight; downward, if the letter did not. The set size of four was selected to maximize task difficulty.

The mapping of flight tasks to UTC-PAB tasks is given in Table 5.

Table 5. Mapping Flight Tasks to UTC-PAB Tasks

Flight Tasks	UTC-PAB Tasks
Responding to an Engine Fire	Visual Memory Search
Completing Checklists	Time Wall Interval Production
Detecting an Oil-Pressure Change	Four-Choice Serial Reaction Time Visual Probability Monitoring Pattern Comparison (Successvie)
Detecting a Heading Discrepancy	Pattern Comparison (Simultaneous)
Controlling the Aircraft	Visual Scanning
Station Keeping	Spatial Processing Matrix Rotation Manikin Test
Aerial Navigation	Mathematical Processing Spatial Processing
Authentication	Code Substitution
Changing Radio Frequencies	Linguistic Processing Grammatical Reasoning (Traditional) Grammatical Reasoning (Symbolic)
Returning from an Empty Channel	Continuous Recognition Auditory Memory Search
Responding to an IFF Query	Four-Choice Serial Reaction Time
Updating the Release Point	Two-Column Addition Mathematical Processing
Dropping Heavy Equipment	Dichotic Listening Unstable Tracking Memory Search Unstable Tracking Combination
Responding to Pilot Commands	Linguistic Processing
Performing a Secondary Task	Visual Memory Search

Note that the Stroop Task from the UTC-PAB was not incorporated into the flight tasks since the Stroop required a color display.

REFERENCES

Angus, R.G. and R.J. Heslegrave, 1985, Effects of Sleep Loss on Sustained Cognitive Performance during a Command and Control Simulation, Behavior Research methods, Instruments and Computers, 17, 55-67.

Baddeley, A.D., 1968, A 3-Minute Reasoning Test Based on Grammatical Transformations, Psychonomic Science, 10, 341-342.

Baddeley, A.D., Cuccaro, W.J., Egstrom, G.H., Weltman, G., and Willis, M.A., 1975, Cognitive Efficiency of Divers Working in Cold Water, Human Factors, 17, 446-454.

Borland, R.G., Brennan, D.H., Nicholson, A.N., and Smith, P.A., 1985, Studies on the possible central and peripheral effects in man of a cholenesterase inhibitor (pyridostigmine). Human Toxicology, 4, 293-300.

Folkard, S., 1975, Diurnal Variation in Logical Reasoning, British Journal of Psychology, 66, 1-8.

Graham, C. and Cook, M.R., 1984, Effects of pyridostigmine bromide on psychomotor and visual performance. AFAMRL-TR-84-052.

Johnson, B.L., Cohen, H.H., Struble, R.P., Setzer, J.V., Anger, W.K., Gutuik, B.D., McDonough, T., and Hauser, P., 1974, Field Evaluation of Carbon Monoxide Exposed Toll Collectors, In C. Xintaras and Johnson, B.L., (Eds.), Behavioral Toxicology (306-328), U.S. Department HEW, Publication Number 74-126, U.S. Government Printing Office, Washington, D.C.

Kay, C.D. and Morrison, J.D., 1985, The effects of a single oral dose of pyridostigmine on contrast sensitivity. Glasgow, U.K.: Institute of Physiology, University of Glasgow.

Ketchum, A.S., Sidell, F.R., Crowell, E.B., Aghajanian, G.K., and Hayes, A.H., 1973, Atropine, Scopolamine, and Ditran: Comparative Pharmacology and Antagonists in Man, Psychopharmacologia, 28, 121-133.

Lewis, J., Baddeley, A.D., Bonham, K.G., and Lovett, D., 1970, Traffic Pollution and Mental Efficiency, Nature, 225, 95-96.

Lohman, D.F., 1979, Spatial Ability: A Review and Reanalysis of the Correlational Literature, (Tech Report Number 8), Stanford, California: Aptitude Research Project, School of Education, Stanford University.

Michelson, M.J., 1961, Pharmacological Evidence of the Role of Acetylcholine in the Higher Nervous Activity of Man and Animals, Activas Nervosa Superior, 3, 140-147.

Michon, J.A., 1966, Tapping Regularity as a Measure of Perceptual Motor Load, Ergonomics, 9, 401-412.

Neisser, V., 1963, Decision-Time Without Reaction-Time: Experiments in Visual Scanning, American Journal of Psychology, 76, 376-385.

Perez, W.A., Masline, P.J., Ramsey, E.G., and Urban, K.E., 1987, Unified tri-service cognitive performance assessment battery: review and methodology. AAMRL-TR-87-007, Wright-Patterson AFB, OH.

Poulton, E.C., Hunt, G.M., Carpenter, A., and Edwards, R.S., 1978, The Performance of Junior Hospital Doctors Following Reduced Sleep and Long Hours of Work, Ergonomics, 21, 279-295.

Repko, J.D., Jones, P.F., Garcia, L.S., Schneider, E.J., Roseman, E., and Corum, C.R., 1976, Behavorial and Neurological Effects of Methyl Chloride, Washington, D.C.: U.S. Government Printing Office, U.S. DHHS (NIOSH) Publication Number 77-125.

Repko, J.D., Morgan Jr., B.B., and Nicholson, J., 1975, Behavioral Effects of Occupational Exposure to Lead, DHEW Publication Number (NIOSH) 75-184, Washington, D.C.: U.S. Government Printing Office.

Schiflett, S., Stranges, S., Slater, T., and Jackson, M., 1987, The effects of pyridostigmine bromide on performance at ground level and altitude. Proceedings of the Aerospace Medical Association Annual Meeting, Las Vegas, Nevada, 1987.

Stroop, J.R., 1935, Studies of Interference in Verbal Reactions, Journal of Experimental Psychology, 18, 643-662.

89-3334-CP

TIME-DOMAIN OPTIMIZATION DESIGN METHOD OF INPUT SIGNALS

Wang Zicai Jia Honghui Zhao Changan
Dept. of Control Engineering
Harbin Institute of Technology, Harbin, China

Abstract

A design problem of optimal input signals for aircraft parameter estimation is developed in this paper . According to the properties of flight test, short testing time, a lot of limitations, high estimation accuracy of parameter and high test cost, for example, a time-domain optimization design method which considers the testing time and accuracy simultaneously is presented.

Using the method presented, an optimal input signal which meet the practical requirements and constrained conditions is obtained and the input signal can be implemented by using multistage design in order to meet the higher requirements for estimated parameter. The simulation results show that the optimal solution exist and has the robustness for initial condition.

Introduction

The main factors to influence the accuracy of parameter estimation are: structure identification method for model, parameter estimation method and experiment design. The main problem of experiment design is to find an optimal input signal. So, the optimal input signal design is a chief problem in the system identification.

Experiments show that the accuracy of parameter estimation can be increased highly by using an optimal input signal to excite the identified system. A lot of researches are done in this line.

The optimal design for input signal is the most important in the parameter estimation of aircraft. The aims of the identification of aircraft are:

a. To provide a correct dynamic model of the aircraft for the aircraft simulation system both in the ground and in the air;

b. To verify the aerodynamic parameters of the aircraft which are got in wind tunnel and theoretical computation;

c. To provide basic data for designing and improving the aircraft control system;

d. To identify the flight performance of the aircraft;

e. To analyze an accident of the aircraft.

Because of high cost for the flight test and the limit of the testing time and condition, it is necessary to find various ways to decrease flight testing time and to increase the accuracy of parameter estimation. Using an optimal input signal, we can get the good results for requirements above.

The common input signals used in identification of aircraft system are: Step signal, Doublet signal, Mehra signal, Schulz signal and so on. It is shown that using the optimal input signal designed by the method suggested in this paper for parameter estimation is better than the Doublet signal and Mehra signal.

Selection of Performance Index

The design problem of an optimal input signal for a continuous system was studied by using Cramer-Rao lower bound and Fisher information matrix. By using maximization of Fisher information matrix, the design problem of an optimal input signal can be transformed to an ordinary integral quadratic optimal control problem with the constraint of input amplitude or energy constraint. Based on these, the design problem of an optimal input signal for identification of a linear system is investigated further by Mehra [1] and a new method for solving the homogeneous two-point boundary-value equations was suggested. Due to the sensitivity equation exists, the order of the system increases, and as we all know that the singular property of Fisher information matrix causes some disadvantages, so the weight of this method is decreased.

For the identification of an aircraft system, it has the characteristic of short time and high accuracy of parameter estimation. The design of an optimal input signal must be based on these characteristics. In common, the order of model of the aircraft is high and there are a lot of unknown parameters, so it is difficulty to use Mehra method and sensitivity method. In order to overcome the singular problem of Fisher information matrix, we can take $\det M$ or $tr\, M^{-1}$ as a performance index. The $tr\, M^{-1}$ is suggested as the performance index in this paper and it will be shown later that the calculation is simplified by using $tr\, M^{-1}$.

As we all know that the high accuracy of parameter estimation means that $C(t_f) = M^{-1}(t_f)$ is small (for an unknown parameter); the short time requirement means that t_f is small. So, in order to design an optimal input signal, the following performance index is suggested

$$J = C(t_f) + t_f \tag{1}$$

where $C(t_f)$ is a covariance matrix. Rewrite equ.(1) as

$$J = tr\, C(t_f) + t_f \tag{2}$$

Considering a weighting coefficient and matrix, equ.(2) can be written as

$$J = tr\, [WM^{-1}] + \alpha t_f \tag{3}$$

where α and W are weighting coefficient and matrix respectively.

Optimization Method in Time Domain

First, a recursion formula for Cramer-Rao lower bound is given, then it can be easily to find the value $tr\, C(t_f)$ by using the formula.

Lemma 1: If the inverse of $A, C, (A+BCD)$ matrices exist, we have the following identity

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}B)^{-1}DA^{-1} \tag{4}$$

The proof is omitted.

For a time-invariant system

$$\dot{x}(t) = Fx(t) + Gu(t) \;,\; x(0) = 0 \tag{5}$$

$$y(t) = Hx(t) + v(t) \tag{6}$$

the Fisher information matrix is of the form

$$M = \int_0^{t_f} x_\theta^T H^T R^{-1} H x_\theta dt \tag{7}$$

where x is $n\times 1$ state vector; u is $m\times 1$ input vector; y is $p\times 1$ output vector; $v(t)$ is a zero-mean Gaussian noise: $E[v(t)]=0$, $E[v(t)v^T(\tau)] = R\,\delta(t-\tau)$; F, G, H are proper dimension constant matrices; θ is an unknown parameter in F; x_θ is the sensitivity function of state variable with respect to θ; i.e., $x_\theta = \partial x/\partial\theta$. The sensitivity equation is

$$\dot{x}_\theta = Fx_\theta + F_\theta x + G_\theta u \tag{8}$$

where $F_\theta = \partial F/\partial\theta$

The integral M can be calculated by

$$M = \sum_{i=1}^{N} x_\theta^T(i) H^T R^{-1} H x_\theta(i) \Delta T \tag{9}$$

$$\Delta T \cdot N = t_f$$

where ΔT is the step length of integral.

Let's $R_1^{-1} = R^{-1}\Delta T$, then

$$M = \sum_{i=1}^{N} x_\theta^T(i) H^T R_1^{-1} H x_\theta(i) \tag{10}$$

Assume

$$M_k = \sum_{i=1}^{k} x_\theta^T(i) H^T R_1^{-1} H x_\theta(i) \tag{11}$$

then

$$M_{k+1} = M_k + x_\theta^T(k+1) H^T R_1^{-1} H x_\theta(k+1) \tag{12}$$

For $C(t_f) = M^{-1}(t_f)$, let's $C_{k+1} = M_{k+1}^{-1}$, then from Lemma 1, we have

$$\begin{aligned} C_{k+1} &= M_{k+1}^{-1} \\ &= [M_k + x_\theta^T(k+1)H^T R_1^{-1} H x_\theta(k+1)]^{-1} \\ &= C_k - C_k x_\theta^T(k+1)H^T [R_1 \\ &\quad + Hx_\theta(k+1)C_k x_\theta^T(k+1)H^T]^{-1} Hx_\theta(k+1)C_k \end{aligned} \tag{13}$$

For SISO system, we have

$$C_{k+1} = C_k - \frac{C_k\, x_\theta^T\,(k+1)H^T\, Hx_\theta\,(k+1)C_k}{R_1 + Hx_\theta\,(k+1)C_k\, x_\theta^T\,(k+1)H^T} \tag{14}$$

Now, we have a conclusion.

Theorem 1. For a system described by the following state, sensitivity and output equations

$$\dot{x} = Fx + Gu \quad, \ x(0) = 0 \tag{15}$$

$$\dot{x}_\theta = F_\theta x + Fx_\theta + G_\theta u \quad, \ x_\theta(0) = 0 \tag{16}$$

$$y = [H,0]\begin{bmatrix} x \\ x_\theta \end{bmatrix} + v(t) \tag{17}$$

and the amplitude of input signal is limited. The design problem of an optimal input signal is to find an input control signal $u(t)$ so that minimizes the performance index

$$J = tr\,[WC(t_f)] + \alpha t_f \tag{18}$$

where $C_{k+1}(t_f)$ can be calculated by the following equation

$$C_{k+1} = C_k - C_k\, x_\theta^T\, H^T\,[R_1 + Hx_\theta\,(k+1)C_k\, x_\theta^T\,(k+1)H^T\,]^{-1}\, Hx_\theta\,(k+1)C_k \tag{19}$$

The optimal algorithm can be summarized as:

1. Given the accuracy ε and variance of parameter estimation, initial value $C(0)$ for recursion calculation, the weighting coefficient α and weighting matrix W;

2. Minimize performance index (18) and then find optimal switching number and time;

3. Check the requirement for accuracy, if the accuracy is met, go next step, otherwise modify W,α and back to the second step;

4. Minimize the performance index $J_2 = tr\; C(t_f)$ to find the switching number and time as the final optimal input signal. The shortest time got from the third step is met the accuracy requirement with the given t_f.

Example

In example, the optimal input signal is found by using the method suggested in this paper for one order system [2], and compared with the Goodwin and Doublet signal.

Given the system

$$\dot{x} = ax + bu \quad, \ x(0) = 0$$

$$y = x + v$$

$$|u| \leq 1$$

where, $E[v(t)] = 0$, $E[v(t)v^T(\tau)] = R\,\delta(t-\tau)$; $R = 10$; a and b are unknown parameters which have priori values $a = -1$, $b = 1$. The sensitivity equation is

$$\dot{x}_a = x + ax_a \quad, \ x_a(0) = 0$$

$$\dot{x}_b = ax_b + u \quad, \ x_b(0) = 0$$

Goodwin use the trace of covariance matrix of the minimum error to get an optimal input signal as shown in Fig.1. Using this input signal the mean square error of the estimated parameter is $\sigma_a = 0.971821$; $\sigma_b = 0.8398283$; $t_f = 10s$. The Doublet signal is shown in Fig.2. Using this input signal, the mean square error of the estimated parameter is $\sigma_a = 1.117046$; $\sigma_b = 0.9257057$; $t_f = 10s$.

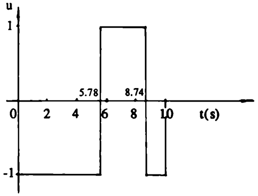


Fig.1. Goodwin signal

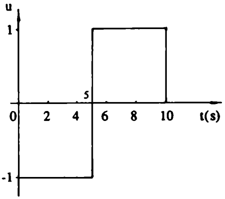


Fig.2. Doublet signal

The optimal input signal by using the method in this paper is shown in Fig.3. Using this signal the mean square error of the estimated parameter is $\sigma_a = 0.9702675$; $\sigma_b = 0.7819366$; $t_f = 9.3s$.

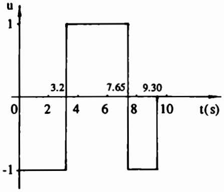


Fig.3. Optimal input signal

The results given above are obtained under the same accùracy for σ_a and σ_b with the Goodwin method. If we consider the σ_a and σ_b together and take $\sigma_a^2 + \sigma_b^2$ as the requirement, then we get the shorter time for the input signal. The mean square error of the estimated parameter is: $\sigma_a = 0.992449$, $\sigma_b = 0.7855733$, $t_f = 9s$. $\sigma_a^2 + \sigma_b^2 = 1.60208 < 1.64975$ (where 1.64975 is the sum of mean square error with the Goodwin signal as an input).

It is shown that the effect of the optimal input is better than of the Goodwin signal, the time of input signal is one second shorter. Because of shorter time and less energy, the cost of data processing is cheaper and the price of experiment is reduced.

The method is also shown advantages for higher order.

Multistage Optimal Design

For a high order system and high accuracy requirement for the estimated parameter, the lower number of switching signal is not met the requirement. It is necessary to find an input signal with higher number. As the switching number increases, the search for optimal switching points is a tedious and difficulty task, and it is difficulty to find an optimal result. In order to overcome the difficulties, the multistage optimal method is suggested. First, given a lower requirement for mean square error, we can easily find an optimal input signal which is the signal with lower switching numbers. Then from the final mean square error and state, according to the reduced mean square error to find the shortest time input signal. If the switching signal with low number can not be met the requirement for accuracy of mean square error, we can do next multistage design for input signal until the requirement for accuracy is met. The input signals obtained in each stage are input signal with low switching number. Connecting these low switching signals obtained for each stage, the input signal is found which is the signal with high switching number and meet the accuracy requirement.

An example of one order system as mentioned above is also given to explain the multistage design method. Let's $\sigma_a = 1.45$, $\sigma_b = 1.15$ as the mean square error requirement for the first stage, the switch input signal can be found as shown in Fig.4 and the results are $\sigma_a = 1.425916$; $\sigma_b = 1.107289$; $x_a = 0.1458945$; $x_b = 0.6926091$.

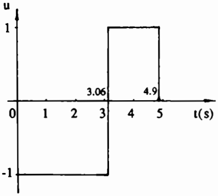


Fig.4. The input signal for the 1'st stage

These mean square errors and states are the initial value for the next stage. Let's $\sigma_a = 1.00$, $\sigma_b = 0.78$ as the mean square error requirement for the second stage, the input signal obtained is shown in Fig.5 and the results under the input signal are $\sigma_a = 0.995195$; $\sigma_b = 0.7754353$; $C(1,1) = 0.991059$; $C(2,2) = 0.6012999$.

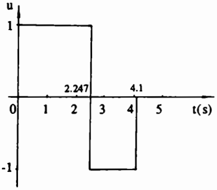


Fig.5. The input signal for 2'd stage

If $\sigma_a = 1.00$, $\sigma_b = 0.78$ are the final requirement for the mean square error, then the design is finished. The final input signal is obtained by connecting the signal in each stage as shown in Fig.6.

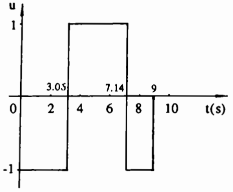


Fig.6. The final input signal

Comparing the signal obtained in Fig.3 with Fig.6, it is shown that the difference between them are smaller, and it is evident that the multistage method is effective and is easy to use.

Example for Application

The optimal input signal design for five unknown parameters of the longitudinal motion of the C-8 airplane is discussed [1]. The comparison in the signal with the signal given in [1] under the same energy of the input is given here to show the advantages of method presented in this paper.

The mathematic model of longitudinal motion of C-8 airplane is of the form

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} M'_q & M'_\alpha \\ 1 & Z_\alpha \end{bmatrix} \begin{bmatrix} q \\ \alpha \end{bmatrix} + \begin{bmatrix} M'_{\delta_e} \\ Z_{\delta_e} \end{bmatrix} \delta_e$$

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} q \\ \alpha \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}$$

where M'_q, M'_α, M'_{δ_e}, Z_α, Z_{δ_e} are five unknown estimated parameters. $v = [v_1, v_2]^T$ is the independent Gaussian white noise, $E[v(t)] = 0$, $E[v(t)v^T(\tau)] = \begin{bmatrix} 0.02 & 0 \\ 0 & 0.04 \end{bmatrix}\delta(t-\tau)$. The initial state is zero. Let's priori value of estimated parameter are $M'_q = -1.588$; $M'_\alpha = -0.562$; $M'_{\delta_e} = -1.66$; $Z_\alpha = 0.737$; $Z_{\delta_e} = 0.005$, and the energy constraint is

$$\int_0^4 u^2(t)dt = 311.$$

Using the design method presented, we can find the input signal as shown in Fig.7. Applying the input signal, the mean square error of estimated parameter are $\sigma_{M'_q} = 0.069738$; $\sigma_{M'_\alpha} = 0.047727$; $\sigma_{M'_{\delta_e}} = 0.039779$; $\sigma_{Z_\alpha} = 0.037445$; $\sigma_{Z_{\delta_e}} = 0.024566$; $tr\ C = \sum_{i=1}^{5} \sigma_i^2 = 0.0107293$.

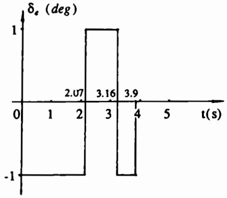


Fig.7. The optimal input signal

The Mehra input signal is shown in Fig.8 and the mean square error of the estimated parameters are $\sigma_{M'_q} = 0.16959$; $\sigma_{M'_\alpha} = 0.06605$; $\sigma_{M'_{\delta_e}} = 0.09601$; $\sigma_{Z_\alpha} = 0.03684$; $\sigma_{Z_{\delta_e}} = 0.02564$; $tr\ C = \sum_{i=1}^{5} \sigma_i^2 = 0.04435$.

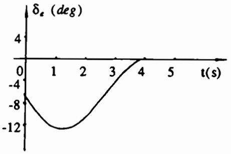


Fig.8. Mehra input signal

The Doublet input signal is shown in Fig.9 and the mean square error of the estimated parameters are $\sigma_{M'_q} = 0.15645$; $\sigma_{M'_\alpha} = 0.28322$; $\sigma_{M'_{\delta_e}} = 0.06398$; $\sigma_{Z_\alpha} = 0.026455$; $\sigma_{Z_{\delta_e}} = 0.06819$; $tr\ C = \sum_{i=1}^{5} \sigma_i^2 = 0.18342$.

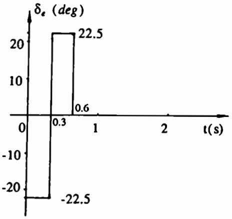


Fig.9. Doublet input signal

Comparing the results given above, it is shown that the design method presented is better than Mehra and Doublet signal. Except for σ_{Z_α}, the mean square error of estimated parameter is reduced in wide range. According to the sum of the variance for five parameters, the accuracy of estimated parameter by using input presented is increased in comparison with the Mehra signal and Doublet signal by 75% and 90% respectively.

Conclusion

The application and simulation results show that the optimal design method in time domain for the optimal input signal is effective, the accuracy for estimated parameter is increased and the computation is simple.

The analysis and simulation verify that the algorithm is always convergence and show that the performance index selected is correct. Under the difference initial conditions, the optimization is convergence to a minimum value of performance index. It is shown that the algorithm has robustness for the initial condition. The design method can overcome the "dimension disaster" by using the method in this papaer, because only $(n+1)m$ order differential equation must be solved. If we use the Mehra method, there are $2m(n+1)$ order differential equation must be solved.

The selection for weighting matrix and coefficient obey some law, so that it can avoid selection the weighting matrix and Lagrange factor in Mehra method. The method for designing input signal given here can also be applied to nonlinear system. So, it is a powerful practical method for designing an optimal input signal in time domain.

Reference

[1] Mehra, R.K., "Optimal Input for System Identification", IEEE Trans. Auto control, AC-19, No 3,1974.

[2] Goodwin, G.C., "Input Synthesis for Minimum Covariance State and Parameter Estimation", Electronic Letters, Vol.5, No 21, 1969.

AIAA Flight Simulation Technologies Conference and Exhibit

AUTHOR INDEX

(Number after name indicates session in which paper appears.)